

**IDENTIFICAÇÃO DE CARACTERÍSTICAS  
RELEVANTES PARA REUTILIZAÇÃO DE  
EXPERIÊNCIAS DE DESENVOLVIMENTO DE  
SOFTWARE**

**Universidade Federal de Santa Catarina**  
**Programa de Pós-Graduação em**  
**Engenharia de Produção**

**IDENTIFICAÇÃO DE CARACTERÍSTICAS  
RELEVANTES PARA REUTILIZAÇÃO DE  
EXPERIÊNCIAS DE DESENVOLVIMENTO DE  
SOFTWARE**

Marcel Pacheco de Souza

Dissertação apresentada ao  
Programa de Pós-Graduação da  
Universidade Federal de Santa Catarina  
como requisito parcial para obtenção  
do título de Mestre em  
Engenharia de Produção

Florianópolis  
2002

Marcel Pacheco de Souza

# IDENTIFICAÇÃO DE CARACTERÍSTICAS RELEVANTES PARA REUTILIZAÇÃO DE EXPERIÊNCIAS DE DESENVOLVIMENTO DE SOFTWARE

Esta dissertação foi julgada e aprovada para a obtenção do título de **Mestre em Engenharia de Produção** no **Programa de Pós-Graduação em Engenharia de Produção** da Universidade Federal de Santa Catarina

Florianópolis, 08 de abril de 2002.

---

Prof. Ricardo Miranda Barcia, Ph.D.  
**Coordenador do Curso**

## BANCA EXAMINADORA

---

Prof<sup>a</sup>. Christiane Gresse von Wangenheim, Dr<sup>a</sup>.  
**Orientadora**

---

Prof. José Eduardo De Lucca, Dr.  
**Membro**

---

Prof. Vinicius Medina Kern, Dr.  
**Membro**

## *Agradecimentos*

A professora Christiane Gresse von Wangenheim pela oportunidade de trabalho concedida, pela orientação sempre prestativa e competente, e pelo tempo e empenho dedicado ao desenvolvimento e conclusão deste trabalho.

Aos professores José Eduardo De Lucca e Vinicius Medina Kern, pela dedicação, comentários e sugestões construtivas ajudando no melhoramento deste trabalho.

Ao grande amigo Lurregi dos Santos Côrrea pelo auxílio, apoio e discussões durante o desenvolvimento do meu trabalho.

Também não poderia deixar de agradecer os professores Mauro Pacheco Ferreira, Lúcia Helena Martins Pacheco e Clara Amélia de Oliveira pela confiança e apoio inicial dados para realização deste projeto.

E, por fim, a todas as pessoas que participaram, direta ou indiretamente, na conclusão deste trabalho.

## Sumário

Lista de Figuras .....	vi
Lista de Quadros .....	vii
Lista de Reduções .....	viii
Resumo .....	ix
Abstract .....	x
1. Introdução .....	1
1.1 Motivação .....	1
1.2 Problema .....	3
1.3 Objetivos .....	4
1.4 Estrutura do Trabalho .....	5
2. Revisão de Literatura .....	6
2.1 Reutilização de Conhecimento .....	6
2.2 Raciocínio Baseado em Casos .....	9
2.2.1 Representação de Casos .....	10
2.2.2 Indexação de Casos .....	15
2.3 Análise de Domínio .....	22
2.3.1 Processo de Análise de Domínio .....	24
2.3.2 Métodos de Análise de Domínio .....	25
3. Metodologia para Indexação de Experiências de Desenvolvimento de Software .....	32
3.1 Etapas do Método .....	32
3.1.1 Etapa 1: Identificação da Meta de Reutilização .....	34
3.1.2 Etapa 2: Caracterização e Estruturação de Casos Avaliáveis .....	37
3.1.3 Etapa 3: Identificação de Características Organizacionais .....	38
3.1.4 Etapa 4: Coleta de Informações sobre Projetos realizados pela Organização .....	39
3.1.5 Etapa 5: Análise e Interpretação dos Dados coletados e Definição de um Conjunto de Índices .....	41
3.1.6 Etapa 6: Monitoramento e Aperfeiçoamento do Conjunto de Índices .....	45

3.1.7 Discussão.....	48
3.2 Ferramenta .....	49
3.2.1 Passo 1: Definição da Meta de Reutilização .....	49
3.2.2 Passo 2: Construção da Base de Conhecimento .....	50
3.2.3 Passo 3: Preenchimento do Questionário Organizacional .....	51
3.2.4 Passo 4: Preenchimento dos Questionários de Projeto .....	52
3.2.5 Passo 5: Definição dos Valores de Relevância .....	53
3.2.6 Passo 6: Relatório e Conjunto Final de Índices.....	55
3.2.7 Discussão.....	56
3.3 Aplicação e Avaliação do Método .....	57
3.3.1 Teste de Aplicação do Método .....	59
4. Conclusão.....	66
5. Fontes Bibliográficas .....	68
6. Anexos.....	72
6.1 Aplicação Experimental da Análise do Domínio para Caracterizar o Desenvolvimento de Software .....	72
6.2 Questionário Organizacional.....	75
6.3 Questionário de Projetos.....	78
6.4 Análise e Projeto da Ferramenta .....	84

## Lista de Figuras

Figura 1: A interação da <i>Experience Factory</i> e da gerência de projetos (fonte [BCR94a]) .....	8
Figura 2: O ciclo de tarefas RBC (fonte [AP90]) .....	10
Figura 3: A aplicação da técnica de análise de domínio para auxiliar o processo de reutilização de software .....	23
Figura 4: Diagrama de contexto SADT para o processo de análise de domínio (fonte: [Dia90]) .....	25
Figura 5: Diagrama do método comum de Arango (fonte: [Ara94]) .....	31
Figura 6: Diagrama mostrando o fluxo de informações da metodologia proposta .....	34
Figura 7: Ferramenta - definição da meta de reutilização .....	50
Figura 8: Ferramenta - construção da base de conhecimento .....	51
Figura 9: Ferramenta - preenchimento do questionário organizacional.....	52
Figura 10: Ferramenta - preenchimento do questionário de projeto.....	53
Figura 11: Ferramenta - definição dos valores de relevância e conjunto final de índices .....	54
Figura 12: Ferramenta - definição de um novo índice (passo 5) .....	55
Figura 13: Ferramenta - tela mostrando o relatório final.....	56

## Lista de Quadros

Quadro 1: Partes de um caso - problema.....	12
Quadro 2: Partes de um caso - causa (problema do quadro 1).....	13
Quadro 3: Partes de um caso - solução (problema do quadro 1).....	14
Quadro 4: Partes de um caso - resultado (problema do quadro 1).....	15
Quadro 5: Etapas e produtos do método FODA.....	27
Quadro 6: Etapas e produtos do método ODM .....	28
Quadro 7: Etapas e produtos do método Synthesis .....	29
Quadro 8: Etapas e produtos do método comum de Arango.....	30
Quadro 9: Estrutura de uma meta de reutilização .....	36
Quadro 10: Estrutura de um caso.....	38
Quadro 11: Algoritmo para definição das características iniciais do questionário organizacional.....	42
Quadro 12: Algoritmo para definição do conjunto inicial de características de projeto.....	43
Quadro 13: Fórmula para o cálculo do peso de cada índice .....	45
Quadro 14: Situação e ação sugerida para aperfeiçoar o conjunto de índices	48
Quadro 15: Teste do método - meta de reutilização selecionada pela organização .....	59
Quadro 16: Teste do método – respostas do questionário organizacional.....	62
Quadro 17: Teste do método – respostas dos questionários de projetos.....	63
Quadro 18: Teste do método – análise comparativa dos questionários .....	64
Quadro 19: Teste do método – nova característica levantada .....	65
Quadro 20: Teste do método – identificação dos valores de relevância .....	65
Quadro 21: Teste do método – conjunto final de características.....	65



## Lista de Reduções

<b>EF</b>	<i>Experience Factory</i>
<b>FODA</b>	<i>Featured-Oriented Domain Analysis</i>
<b>GQM</b>	<i>Goal/Question/Metric approach</i>
<b>HTA</b>	<i>Hierarchical Task Analysis</i>
<b>IA</b>	Inteligência Artificial
<b>ODM</b>	<i>Organization Domain Modeling</i>
<b>QIP</b>	<i>Quality Improvement Paradigm</i>
<b>RBC</b>	Raciocínio Baseado em Casos
<b>SEI</b>	<i>Software Engineering Institute</i>
<b>SEL</b>	<i>Software Engineering Laboratory</i>
<b>SPC</b>	<i>Software Productivity Consortium</i>

## Resumo

A captura e a reutilização de conhecimento, adquirido através do desenvolvimento de software e ajustado às características e necessidades específicas de cada organização, são fatores importantes para assegurar o aumento de produtividade e a melhoria contínua de qualidade no desenvolvimento de projetos de software. Neste sentido, deve-se considerar que o conhecimento relevante no desenvolvimento de software difere entre organizações, assim como os projetos desenvolvidos por uma mesma organização possuem características que os distinguem conforme seu domínio de aplicação. Por isso, a caracterização do contexto do conhecimento, obtido através da utilização de técnicas de engenharia de software, deve levar em consideração as características específicas de cada organização e o ambiente no qual o projeto foi desenvolvido. A representação destas características pode ser feita através de índices que têm como finalidade facilitar o processo de recuperação e reutilização de conhecimento. Este trabalho apresenta uma metodologia e uma ferramenta computacional que permite identificar e representar o contexto do conhecimento obtido no desenvolvimento de projetos de software através da determinação de características que descrevem um conjunto de experiências produzidas durante o desenvolvimento de sistemas de software.

**Palavras-chave:** reutilização de conhecimento, raciocínio baseado em casos, qualidade de software e análise de domínio.

## Abstract

Capture and reuse of explicit software engineering know-how, acquired during software development and tailored to specific characteristics and needs of a particular organization, are essential issues to assure the continuous improvement of productivity and quality in software projects. However, it must be considered that relevant knowledge differs among companies and many software projects developed by the same company may have features that make them completely different depending on their application domain. Thus, particular definition of knowledge, acquired through the usage of software engineering techniques, must consider specific characteristics of each organization and context where projects were done. Indexes, used to improve knowledge reuse and recover, can help to represent these features. This dissertation introduces a methodology and a computational tool that permits to represent the context of knowledge acquired during the development of software projects through the identification of contextual characteristics to represent a set of software systems experiences.

**Key words:** knowledge reuse, case based reasoning, software quality and domain analysis.

# 1. INTRODUÇÃO

## 1.1 Motivação

Atualmente os sistemas computacionais são utilizados nas mais diversas áreas e possuem as mais variadas finalidades. A utilidade e importância dos sistemas computacionais têm gerado uma necessidade cada vez maior para que sejam desenvolvidos produtos de software com alto nível de qualidade e perfeição. Devido à grande demanda estabelecida na indústria de software e a exigência de sistemas cada vez mais complexos e sofisticados, o desenvolvimento de projetos de software precisa empregar paradigmas de desenvolvimento que priorizem a construção de sistemas bem planejados, estruturados, documentados, com custos e objetivos bem definidos. Neste contexto, o QIP (*Quality Improvement Paradigm*) [BCR94a, Wan98], desenvolvido em conjunto pela universidade de Maryland e pelo SEL (*Software Engineering Laboratory*), apresenta-se como uma solução bastante apropriada.

O QIP é uma abordagem de desenvolvimento de software que tem como meta melhorar a qualidade dos processos e produtos de software baseado no aprimoramento contínuo e sistemático do desenvolvimento de software. O seu funcionamento é fundamentado na identificação de pontos fortes e fracos de processos e produtos de software no seu ambiente específico de desenvolvimento. Através da obtenção de dados quantitativos e qualitativos, pontos potenciais de melhoramento são determinados e ações são propostas. A metodologia QIP baseia-se em três aspectos principais que são: um programa de mensuração orientado a metas; a modelagem explícita de produtos e processos; e a reutilização de conhecimento através do armazenamento, recuperação e adaptação de experiências adquiridas.

O QIP utiliza um programa de mensuração orientado a metas para compreender e avaliar os processos e produtos de software, fornecer suporte ao planejamento de novos projetos, controlar o andamento dos projetos atuais

e também auxiliar o melhoramento contínuo do desenvolvimento de software. A aquisição de dados qualitativos e quantitativos, através de programas de mensuração, permite identificar dados importantes na análise, controle ou avaliação de um projeto. Desta forma é possível levantar informações valiosas, sobre o desenvolvimento de software, previamente desconhecidas ou definidas de maneira informal.

De forma a aprimorar o conhecimento adquirido através do uso de programas de mensuração, ou qualquer outro artefato ou produto obtido durante o desenvolvimento de software, memórias corporativas precisam ser utilizadas para facilitar e gerenciar a coleta e a reutilização do conhecimento adquirido. A EF (*Experience Factory*) [BCR94a, WRW<sup>+</sup>99] é um modelo de memória corporativa. Ela apresenta uma organização lógica e física que apóia o desenvolvimento de projetos analisando e sintetizando todos os tipos de experiência, atuando como repositório para elas, e fornecendo conhecimento, através de um conjunto de casos, para vários projetos em demanda. [BCR94a]. Para operacionalizar a estrutura fornecida pela EF, tecnologias que permitam a captura, representação e armazenamento de experiências, além de possibilitarem o reuso eficiente e efetivo do conhecimento em novos projetos, precisam ser implementadas. Portanto, um sistema baseado em conhecimento precisa ser utilizado como uma plataforma de suporte integrado à EF. Neste contexto, a abordagem RBC (Raciocínio Baseado em Casos) [Hen97, AP94, ABT98, Kol93, Wat97] fornece o suporte necessário para a construção e utilização de sistemas baseados em conhecimento.

O RBC é um enfoque de aprendizado contínuo e incremental que utiliza o conhecimento de situações concretas acontecidas no passado (casos<sup>1</sup>) para resolver novos problemas e aprender através da experiência adquirida. O principal objetivo da abordagem RBC é o desenvolvimento de uma base de experiências na prática e o fornecimento de um suporte amplo ao

---

<sup>1</sup> Um caso, em RBC, é uma forma de conhecimento contextualizado representando uma experiência que ensina uma lição útil [Kol93].

desenvolvimento de sistemas baseado nesta base de conhecimento [WWB98]. Em um processo RBC, a descrição inicial de um problema define um novo caso. Este novo caso é utilizado para recuperar um ou mais casos da coleção de casos passados que estão armazenados na base de experiências. O caso recuperado é combinado com o novo caso, e uma solução inicial é proposta com base no caso inicial. A solução proposta é aplicada e através de um processo de revisão o novo caso é avaliado [AP94]. A avaliação segue da seguinte forma: quando um problema é resolvido com sucesso, a experiência é guardada de modo a resolver problemas similares posteriormente. Quando uma tentativa de resolver um problema falha, a razão da falha é identificada e lembrada de forma a evitar o mesmo erro no futuro. Depois de analisada, a nova experiência é armazenada e a base de casos é atualizada por um novo caso aprendido, ou pela modificação de algum caso já existente.

Um sistema baseado em RBC é extremamente dependente da estrutura e do conteúdo contido na sua base de casos. Os processos de representação e recuperação precisam ser ambos efetivos e eficientes de modo a tornar um sistema RBC útil. Cada vez que um novo caso é adquirido ele tem de ser armazenado de algum modo. O problema de representação de conhecimento em RBC está em decidir o que armazenar em um caso, encontrar uma estrutura apropriada para descrever o conteúdo dos casos, e decidir como uma base de casos deveria ser organizada e indexada para uma efetiva recuperação e reutilização do conhecimento contido nela.

## **1.2 Problema**

O conhecimento adquirido em um projeto de software, através da utilização de técnicas de Engenharia de Software, pode ser melhor aproveitado através de sua transferência para outros projetos de uma mesma organização. A disseminação das experiências aprendidas permite o melhoramento contínuo do desenvolvimento de projetos de software e também proporciona a

diminuição de seus custos. Porém, para que a transferência de conhecimento seja possível e viável, é preciso que as experiências adquiridas sejam catalogadas e armazenadas de modo que elas não sejam futuramente esquecidas ou perdidas. Além disso, é necessária a identificação de características de cada experiência e do projeto no qual ela se originou para que este conhecimento possa ser transferido entre projetos similares.

A determinação de índices<sup>2</sup>, que representem o contexto onde a experiência foi produzida, torna o processo de recuperação de conhecimento mais consistente. Os índices permitem a comparação entre os contextos onde as experiências se originaram, sendo possível assim determinar a similaridade contextual entre elas. Como consequência, a reutilização de conhecimento tende a tornar-se mais efetiva, porque os casos recuperados serão semelhantes aos problemas enfrentados.

Para definir sistematicamente um conjunto de índices que identifiquem e representem uma base de conhecimento é importante a utilização de guias que auxiliem o processo de seleção destes índices. Estes guias são utilizados para auxiliar a representação do conjunto de experiências contido em uma base de conhecimento e deste modo proporcionar uma melhor recuperação e reutilização do conhecimento.

### 1.3 Objetivos

O principal objetivo deste trabalho é a elaboração de uma metodologia que auxilie o processo de escolha de características contextuais que irão representar um conjunto de experiências obtido durante o desenvolvimento de sistemas de software.

---

<sup>2</sup> Índice é um rótulo que associado a um valor específico é utilizado de forma a caracterizar e posteriormente recuperar casos similares em uma base de casos.

Outro objetivo desta dissertação é o desenvolvimento de uma ferramenta computacional para facilitar a aplicação da metodologia na prática. Com este intuito, são fornecidos recursos para auxiliar a aquisição e o armazenamento de dados e mecanismos para automatizar os processos de determinação de índices.

O terceiro objetivo deste trabalho é a aplicação do método em um ambiente empresarial. Os dados obtidos nesta avaliação são analisados, e uma avaliação inicial sobre o comportamento do método e os resultados obtidos é realizada.

## **1.4 Estrutura do Trabalho**

Este trabalho está estruturado em seis capítulos. O capítulo dois apresenta os conceitos de abordagens e processos utilizados no desenvolvimento desta dissertação: a reutilização de experiências de desenvolvimento de software, a técnica RBC e a técnica de análise de domínio. O terceiro capítulo descreve a metodologia desenvolvida para caracterização de experiências de desenvolvimento de software e a ferramenta construída para suportá-la. Finalizando este trabalho estão as conclusões (capítulo quatro), as referências bibliográficas (capítulo cinco) e os anexos.



## **2. REVISÃO DE LITERATURA**

Neste capítulo, são apresentados algumas técnicas e processos que serviram como base teórica para o desenvolvimento deste trabalho. O processo de reutilização de conhecimento, a técnica de RBC, e o enfoque de análise de domínio são introduzidos.

### **2.1 Reutilização de Conhecimento**

Nas últimas décadas, a produtividade no desenvolvimento de software tem melhorado constantemente, porém ela ainda não alcançou um nível suficiente para atender a demanda imposta à indústria de software. Apesar do grande progresso atingido com relação à compreensão dos mecanismos de construção de um programa desde sua especificação até a implementação de seu produto final, muito pouco progresso foi obtido no melhoramento das práticas de desenvolvimento de software. Como consequência, existe uma grande diferença entre a demanda do mercado (em uma sociedade que é cada vez mais dependente de software e cada vez menos tolerante a falhas) e o que a indústria de software realmente pode produzir. Em razão disto, após anos de intensiva pesquisa em Engenharia de Software e IA (Inteligência Artificial), chegou-se à conclusão que a reutilização de experiências de software é uma (e talvez a única) alternativa que proporciona a melhoria de produtividade e qualidade que a indústria de software tanto precisa [MMM95].

Atualmente diversos fatores retardam a reutilização de software, incluindo o pouco tempo de vida da Engenharia de Software como uma ciência, o inadequado treinamento de desenvolvedores no processo de desenvolvimento de software como um todo e da reutilização em particular, as estruturas e práticas inadequadas de gerenciamento de projetos de software, e a falta de metodologias e ferramentas que suportem a reutilização de software

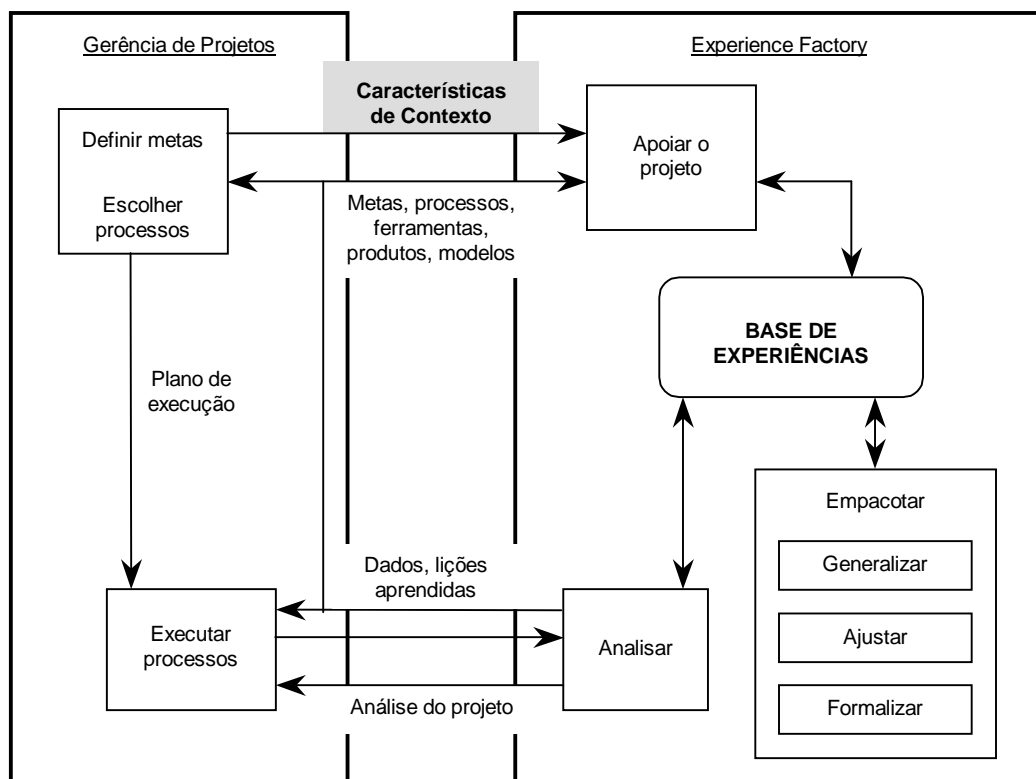
em específico e o desenvolvimento de software em geral. Por isso, a utilização de abordagens de Engenharia de Software que incluam a reutilização de conhecimento pode tornar o desenvolvimento de software cada vez mais eficiente, tornando possível assim o aumento na produtividade e na qualidade de software que as organizações tanto necessitam.

Tradicionalmente, os desenvolvedores pensam sobre reutilização, com a idéia de reutilizar somente objetos concretos, principalmente “código fonte”. Contudo, não apenas os produtos gerados durante o processo de desenvolvimento de software podem ser reutilizados como também todo tipo de experiência adquirida durante este processo, tais como: lições úteis obtidas através da aplicação de determinados métodos e, modelos de produtividade obtidos no desenrolar do processo de desenvolvimento. Normalmente, a experiência obtida durante o desenvolvimento de software é armazenada de forma implícita e dificilmente está acessível. Este fato acontece devido à falta de uma infra-estrutura que permita que o conhecimento adquirido não seja perdido e que forneça meios dele circular dentro da organização [Hof97]. Uma tecnologia avançada neste propósito é a *Experience Factory*.

A *Experience Factory* é uma organização lógica e física que apóia o desenvolvimento de projetos pela análise e síntese de todo tipo de experiência, servindo como repositório para experiências, e fornecendo conhecimento para vários projetos em demanda [BCR94a]. O gerenciamento de projetos e a *Experience Factory* são organizações separadas que se comunicam entre si, como mostra a figura 1. A tarefa da gerência do projeto é desenvolver e manter o software. Ela determina as metas e características que são necessárias para obter-se *feedback* da *Experience Factory* em forma de modelos de conhecimento que se ajustam às metas e características dadas e que são úteis para resolver o problema atual. Depois que os modelos de experiências são reutilizados em um projeto, a gerência de projeto realimenta a *Experience Factory* com novas experiências (lições aprendidas).

A parte central da *Experience Factory* é sua base de experiências. Uma base de experiências efetiva contém um conjunto acessível e integrado de

modelos de experiência analisado, sintetizado e empacotado que descreve experiências do passado, de forma a apoiar o desenvolvimento de software.



**Figura 1: A interação da *Experience Factory* e da gerência de projetos (fonte [BCR94a])**

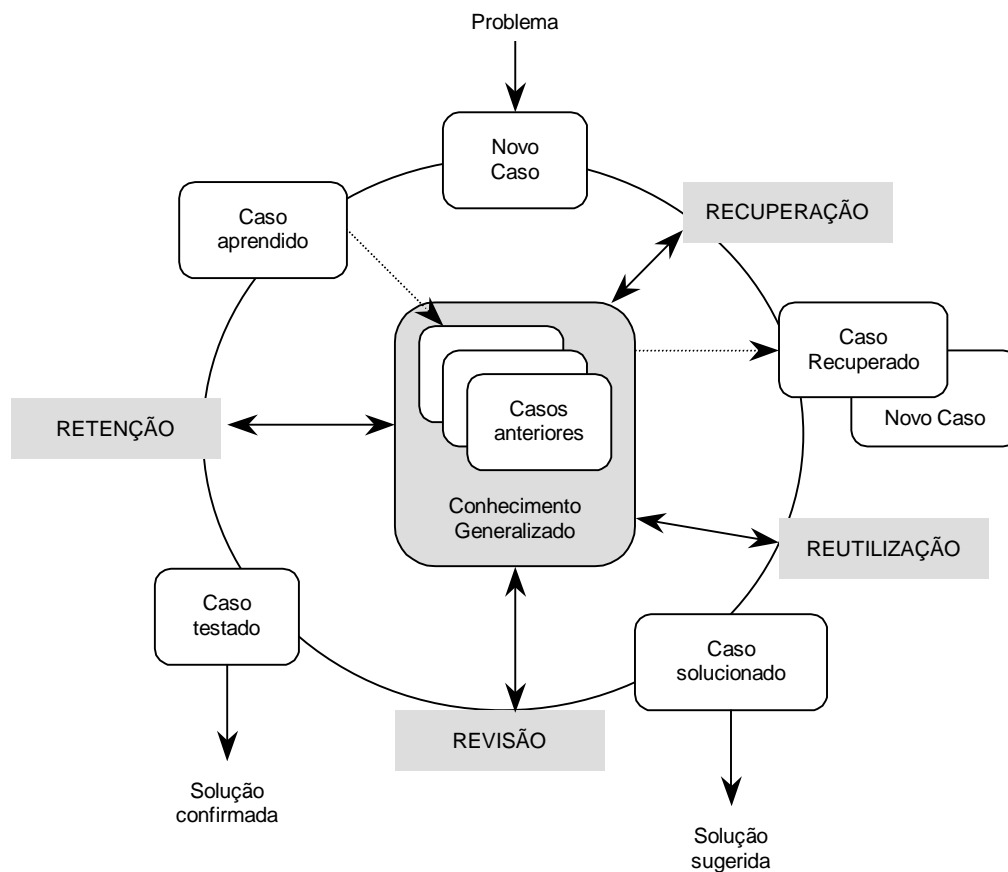
Em uma forma mais abrangente, o problema da reutilização de software não se encontra somente na operacionalização de uma estrutura que apóie a reutilização, mas também reside no fato que o desenvolvimento de software não é um campo uniforme, e a necessidade por diversas soluções computacionais nas mais diferentes áreas produzem aplicações com características bastante distintas [Hen95]. Por isso, para reutilizar o conhecimento obtido durante o processo de desenvolvimento de software em geral são necessárias a identificação e a representação do ambiente específico de desenvolvimento de software onde as experiências se originaram. A eficácia do processo de reutilização depende muito da situação onde ele é aplicado. Geralmente, projetos de software possuem diferenças conforme sua área de aplicação, seus objetivos, seu contexto, por isso para que o processo de reutilização apresente sucesso, é extremamente importante concentrar

esforços na reutilização de software em famílias de aplicações com características similares.

## **2.2 Raciocínio Baseado em Casos**

RBC (Raciocínio Baseado em Casos) é um paradigma para resolução de problemas que utiliza o conhecimento específico de experiências passadas para solucionar situações de problemas concretos. Ao contrário de alguns métodos de Inteligência Artificial que criam associações sobre relacionamentos generalizados entre descrições de problemas e soluções, em RBC um problema é resolvido através da obtenção de um caso similar no passado e da sua reutilização em um novo problema. O método também utiliza uma abordagem para aprendizagem sustentada e incremental, uma vez que uma nova experiência é armazenada cada vez que um problema é resolvido, tornando-se disponível para problemas futuros [Wan00]. A aprendizagem é uma característica muito importante no enfoque RBC, porém a aprendizagem efetiva requer um processo bem trabalhado de forma a extrair conhecimento relevante da experiência, integrar o caso em uma estrutura de conhecimento já existente, e indexar o caso para futura recuperação em casos similares [AP94].

O processo de funcionamento da técnica RBC, mostrado na figura 2, gira em torno da resolução de novos problemas e da aprendizagem pela experiência adquirida. As principais tarefas que compõe o ciclo RBC são: a descrição de uma situação de problema atual (representação e indexação), a procura por um caso no passado que seja similar ao novo caso (recuperação), a utilização deste caso para sugerir uma solução para o problema atual (reutilização), a avaliação da solução proposta (revisão e adaptação) e a atualização do sistema pela aprendizagem desta nova experiência (armazenamento e aprendizagem).



**Figura 2: O ciclo de tarefas RBC (fonte [AP90])**

Nos próximos sub-capítulos os tópicos de representação e indexação de casos são tratados com maior profundidade por estarem fortemente relacionados ao trabalho desenvolvido. Informações mais detalhadas sobre os outros fundamentos (recuperação, reutilização, adaptação, armazenamento e aprendizagem) podem ser encontradas em [Kol93, AP94, Wat97].

### 2.2.1 Representação de Casos

Um caso é pedaço de conhecimento contextualizado representando uma experiência [Wat97]. Cada caso possui duas partes principais: as lições que ele ensina (conteúdo) e o contexto no qual estas lições podem ser utilizadas

(índices contextuais). O conteúdo de um caso é constituído tipicamente de quatro partes:

- Problema: descreve a situação no momento em que o caso aconteceu e, se apropriado, o que é necessário para sua resolução;
- Causa: descreve os fatores que causaram a ocorrência do problema;
- Solução: propõe um conjunto de ações a serem executadas para solucionar o problema descrito;
- Resultado: descreve o estado resultante quando a solução foi aplicada.

Os componentes de um caso constituem uma estrutura útil para sua representação. Eles também fornecem meios para especificar que tipo de conhecimento é necessário para que diferentes tarefas sejam realizadas. Por exemplo, casos que incluem uma solução podem ser utilizados para derivar soluções para novos problemas. Aqueles que possuem o resultado de uma solução aplicada servem para avaliação de novas situações e antecipação de problemas potenciais antes deles ocorrerem. Nas próximas seções, as partes do conteúdo de um caso são melhor detalhadas.

#### **2.2.1.1 Problema**

A descrição de problema representa uma situação que precisa ser resolvida ou uma situação que precisa ser interpretada, classificada ou entendida. Geralmente, um caso passado é apropriado para resolver uma nova situação quando as descrições do problema atual e do problema contido na base de experiência são similares. Deste modo a representação do problema deve ser suficientemente detalhada para que seja possível julgar sua

aplicabilidade em uma nova situação [Kol93]. O quadro 1 apresenta como exemplo a descrição de um problema. Os componentes de um problema são:

- Descrição: descreve o problema propriamente dito;
- Metas: são objetivos concretos ou abstratos a serem alcançadas através da resolução do problema;
- Restrições: são condições aplicadas às metas;
- Características do problema e a relação entre suas partes: é a informação descritiva sobre qualquer situação relevante que exerce influência sobre as metas propostas.

Problema	
Descrição	A empresa “Z” deseja melhorar o visual gráfico de seu site na internet.
Meta	Melhorar o visual gráfico de páginas na <i>web</i> .
Restrições	O investimento financeiro não deve ultrapassar o valor “X”.
Características do problema	No decorrer deste projeto, alguns desenvolvedores serão remanejados de outros projetos, assim que concluírem seus trabalhos atuais, para auxiliar na elaboração do novo projeto gráfico do site.

**Quadro 1: Partes de um caso - problema**

### 2.2.1.2 Causa

A causa descreve os fatores que deram origem ao problema. Ela é composta por uma descrição e pelas circunstâncias na qual o problema ocorreu. As circunstâncias representam o estado ou a situação que gerou o problema. O quadro 2 traz a causa do problema descrito no quadro 1.

Causa	
Descrição	A <i>homepage</i> atual da empresa “Z” possui baixa qualidade gráfica.
Circunstâncias	Quando o site atual foi produzido, havia a necessidade do desenvolvimento em curto prazo das páginas <i>web</i> que comporiam o site da empresa por esta razão naquela ocasião não foi dada ênfase ao aspecto gráfico, mas sim a funcionalidade do site. Além disso, no momento em que as páginas foram construídas, poucos profissionais qualificados estavam envolvidos na tarefa do projeto gráfico do site.

**Quadro 2: Partes de um caso - causa (problema do quadro 1)**

### 2.2.1.3 Solução

A solução constitui-se dos conceitos ou objetos que resolvem o problema, levando em consideração as restrições previstas e outros aspectos contextuais especificados [Kol93]. No quadro 3, encontra-se a solução proposta ao problema descrito no quadro 1. A solução, normalmente é composta por:

- Descrição: é a solução propriamente dita;
- Passos: conjunto de passos utilizados para resolver o problema;
- Justificativas: conjunto de justificativas para decisões tomadas na resolução do problema;
- Soluções Alternativas: conjunto de soluções alternativas não escolhidas (e as justificativas do porquê não foram selecionadas);
- Soluções não aceitáveis: soluções que foram excluídas por não serem adequadas (e as justificativas por sua exclusão);
- Expectativas: expectativas sobre a aplicação da solução.



Solução	
Descrição	Aprimorar o desenho gráfico das páginas utilizando para isso ferramentas computacionais para edição multimídia de imagem, som e vídeo.
Passos	1. Comprar licenças para os softwares de edição multimídia; 2. Fornecer treinamento aos designers para manusear as ferramentas adquiridas.
Justificativas	O investimento em termos de custo não irá ultrapassar o valor pré-determinado.
Soluções alternativas	Terceirizar o serviço contratando uma outra empresa para realizar a tarefa. Justificativa: Não foi utilizada, pois se deu preferência ao aprimoramento técnico da equipe de desenvolvedores através do conhecimento adquirido pela utilização de novas tecnologias e ferramentas.
Soluções não aceitáveis	Compra de novos equipamentos, software para produção das páginas, e incremento no número de pessoas na equipe técnica. Justificativa: o custo para aplicação era superior ao valor pré-estipulado.
Expectativas	Construção de páginas com melhor qualidade visual.

**Quadro 3: Partes de um caso - solução (problema do quadro 1)**

#### 2.2.1.4 Resultado

O resultado especifica o que aconteceu pela aplicação da solução ou como a solução proposta se comportou. O resultado do problema descrito no quadro 1 é mostrado no quadro 4. As partes do resultado são compostas de:

- Descrição: o resultado propriamente dito;
- Expectativas: informa se o resultado preencheu ou violou as expectativas;
- Aplicação da solução: indica se a aplicação da solução resultou em sucesso ou apresentou falha;
- Estratégias: representam as estratégias sugeridas para evitar o problema;
- Explicação para falha: é a explicação dada caso uma falha tenha ocorrido;
- Estratégia de reparo: é a estratégia utilizada no caso do resultado obtido ter apresentado falha;

- Nova solução: é a próxima solução possível apontada pelo resultado da aplicação do reparo.

Resultado	
Descrição	A solução aplicada resultou em novas páginas com maior atrativo visual. Outro fator importante a ressaltar foi que o número de acessos a homepage da organização duplicou nos três primeiros meses após seu lançamento.
Expectativa	O resultado final agradou tanto aos patrocinadores do projeto quanto aos usuários finais.
Aplicação	Sucesso
Estratégia para evitar o problema	Manutenção constante do aspecto gráfico das páginas do <i>site</i> .

**Quadro 4: Partes de um caso - resultado (problema do quadro 1)**

### 2.2.2 Indexação de Casos

Índices são combinações de importantes características em um caso, que o distinguem dentre outros casos e informam sobre quais circunstâncias este caso trará uma lição útil para ensinar e, portanto quando ele poderá ser utilizado. O problema da indexação de casos em RBC divide-se em duas partes. A primeira delas é a associação de rótulos para os casos de forma a identificá-los e assegurar que os casos apropriados serão recuperados corretamente. Neste sentido, as tarefas realizadas e os domínios compreendidos pelos casos precisam ser analisados para encontrar descritores relevantes que deverão ser utilizados para caracterizar os casos. O segundo, é o problema da organização de casos de modo que a procura através da base de experiências seja realizada de forma eficiente e precisa. Neste caso, algoritmos de recuperação devem ser construídos e sua eficiência deve ser testada. O presente trabalho preocupa-se com o primeiro problema.

O conjunto de índices escolhidos para representar um caso deve conter a combinação de características deste caso que foi responsável por sua falha ou pelo seu sucesso. Assim, casos recuperados através da descrição de

características responsáveis por sua falha poderiam ser utilizados para ajudar o raciocinador <sup>3</sup> a antecipar e evitar problemas que aconteceram no passado ou ajudá-lo a criticar a solução. Casos recuperados através da descrição de características responsáveis pela aplicação de uma solução com sucesso seriam úteis na sugestão desta solução em uma nova situação. É importante ressaltar que este enfoque é funcional e pragmático, as tarefas que o caso deveria ser utilizado são analisadas e as características que o descrevem são escolhidas quando ele for útil para estas tarefas. Este enfoque é completamente diferente do enfoque tradicionalmente utilizado na indexação de banco de dados. No caso dos bancos de dados, as características escolhidas como índices são aquelas que, geralmente, produzem uma melhor distribuição da estrutura organizacional do banco. Já em RBC, índices são utilizados para distinguir um caso de outro por determinado propósito, não existe, necessariamente, a necessidade do balanceamento na estrutura da base de casos [AP94].

A indexação não é uma tarefa simples. Para construir uma boa coleção de índices para um conjunto de casos é necessário ter em mente o que faz um índice “bom” e como escolhê-lo. A comunidade de pesquisadores RBC, através da análise de lembranças (enfoque explicado a seguir nesta seção) aliado à experiência adquirida na construção de sistemas RBC, tem proposto um conjunto de características para a escolha de “bons” índices:

- Índices devem ser preditivos. As características preditivas são aquelas combinações de descritores de um caso responsáveis por sua solução, ou influenciam seu resultado;
- Índices devem ser úteis. Índices úteis são aqueles que rotulam um caso de forma a auxiliar nas decisões que o raciocinador irá enfrentar.

---

<sup>3</sup> Raciocinador é a pessoa que utiliza o sistema de conhecimento RBC para encontrar soluções a um determinado problema e que posteriormente pode reutilizar a solução proposta. A palavra raciocinador foi traduzida da palavra “*reasoner*” em inglês.

- Índices devem ser abstratos o suficiente para tornar o caso útil em uma variedade de futuras situações. Índices devem ser mais abstratos do que os detalhes de um caso particular. Embora casos sejam específicos, índices abstratos para casos precisam ser escolhidos para que o caso possa ser usado amplamente em um conjunto de situações apropriadas.
- Índices devem ser concretos o suficiente para serem facilmente reconhecíveis em futuras situações. Embora os índices precisem ser abstratos para cobrirem uma grande variedade de casos, eles, além disso, precisam ser concretos o suficiente para que possam ser reconhecidos com pouca inferência.

Os índices de uma base de experiências são representados através de um vocabulário. O vocabulário é uma estrutura composta por: um conjunto de dimensões coberto por ele, e um conjunto de símbolos utilizado na descrição de determinada dimensão. As dimensões são características, e os símbolos são os valores definidos para determinada dimensão. Os símbolos e dimensões em RBC funcionam estruturalmente da mesma maneira que os atributos e valores na representação de objetos nas linguagens de especificação de sistemas orientadas a objetos. Atualmente, existem dois enfoques principais para determinar quais dimensões e símbolos utilizar na construção do vocabulário de indexação:

- Enfoque funcional: através da análise de um conjunto de casos avaliáveis, determina-se o propósito de cada caso e as formas que ele pode ser descrito para torná-lo disponível;
- Enfoque por lembranças (*reminding approach*): através da análise dos tipos de lembranças comuns entre especialistas que desenvolvem determinada tarefa, características similares entre novas situações que os especialistas se defrontam e os casos que eles relembram são utilizados de modo a

determinar quais tipos de descritores são importantes pelo julgamento de similaridade e em quais circunstâncias.

A utilização destes enfoques irá produzir como resultado a definição de quais dimensões são importantes, quais faixas de valores devem ser empregados e qual nível de detalhes que deve ser usado para obter uma boa representação de uma base de conhecimento. Porém, obter um vocabulário que cubra praticamente todo o domínio representado pelos casos avaliáveis é uma tarefa praticamente impossível, além de que não existe nenhuma forma de provar que o vocabulário definido está correto. Por isso, cabe ao responsável pela construção e manutenção do sistema RBC a tarefa de verificar se o vocabulário criado é suficientemente intuitivo, tem expressividade considerável e suporta um grande conjunto de lembranças.

### **2.2.2.1 Métodos para Seleção de Índices**

Os métodos de seleção de índices são aplicados para determinar qual conjunto de características é necessário para indexar uma base de conhecimento. Atualmente quatro métodos são freqüentemente utilizados no processo de seleção de índices, todos eles apresentam pontos fortes e fracos que devem ser analisados e avaliados conforme sua aplicabilidade, complexidade e desempenho. Os métodos pesquisados são mais bem detalhados em [Kol93]. A seguir cada um dos métodos é introduzido.

#### **2.2.2.1.1 Método *Checklist***

O método *Checklist* escolhe seus índices através da determinação de uma lista de características relevantes que são selecionadas tomando em consideração as tarefas realizadas pelos casos e o domínio dos casos

avaliáveis. Todos os casos são indexados por um conjunto fixo de dimensões selecionado na lista. Este método tem como grande vantagem ser computacionalmente simples e não necessitar de muito conhecimento sobre cada caso para ser aplicado, porém apresenta como pontos negativos: a falta de garantia que todos os índices escolhidos são realmente preditivos, e a possibilidade do conjunto de índices gerado ser muito extenso. Ele é baseado na filosofia que diz que “a determinação de índices corretos é computacionalmente difícil, por isso faz mais sentido analisar o domínio e as tarefas de cada caso e descobrir quais características ou combinação de características tendem a ser preditivas”. A construção da lista é feita pelo responsável pela construção do sistema RBC, seguindo as seguintes etapas:

1. Determina-se quais tarefas o sistema deve suportar, através da análise de seu objetivo;
2. Para cada uma das tarefas, deve-se determinar quais características que têm a tendência de ser importantes através das inferências que apóiam estas tarefas;
3. Um vocabulário apropriado, baseado nas tarefas determinadas, deve ser escolhido para compor a lista.

#### **2.2.2.1.2 Método *Difference-Based***

O método *difference-based* concentra-se na extração de diferenças entre um novo caso e os casos contidos na base de experiências. Na prática este método é usualmente combinado ao método de *checklist* para obter um conjunto de características que assegure um melhor grau de predição e distintividade entre os casos. Assim como o método *checklist*, este método é computacionalmente simples e requer pouco conhecimento sobre o domínio da base de casos, porém não há garantia da escolha de índices preditivos para cada caso individualmente. O método *difference-based* combinado ao método *checklist* é realizado através dos seguintes passos:

1. Para cada caso, seleciona-se tipos de características que sejam reconhecidas como preditivas. Estas dimensões são índices potenciais.
2. Para cada característica, computa-se seu valor para este caso (resultando em pares dimensão-valor);
3. Através da análise do contexto dos casos, retiram-se da lista as características não preditivas;
4. Retira-se da lista tipos de características repetidas.

#### **2.2.2.1.3 Método *Explanation-Based***

O método *explanation-based* utiliza-se de explicações do porquê uma solução funcionou com sucesso ou do porquê falhou para a escolha dos índices. A maior vantagem deste método é que ele permite que melhores identificadores sejam associados com os casos do que os dois métodos anteriores, porém ele é bastante dependente do conhecimento contido em cada caso, além de possuir alta complexidade computacional necessitando que a máquina derive explicações e alcance conclusões. Sua filosofia diz que “os casos deveriam ser analisados individualmente conforme seu resultado (sucesso ou falha) e os índices deveriam ser associados de forma precisa para cada caso em particular”. A seleção de índices por este método requer grande quantidade de conhecimento, tornando-o proibitivo em grandes domínios. O processo de seleção *explanation-based* é composto dos seguintes passos:

1. Determina-se uma explicação;
2. Selecionam-se características relevantes observáveis na explicação;
3. Generaliza-se estas características tanto quanto possível, de forma que a explicação original continue sendo aplicável;
4. Se o índice apoiar a obtenção de uma solução então:

- Adiciona-se informação específica da solução atingida;
- Generaliza-se a solução e, caso a solução seja composta de um conjunto de tarefas específicas, repete-se o processo para cada tarefa.

#### **2.2.2.1.4 Método Manual**

O método de seleção manual de índices é geralmente utilizado quando a representação dos casos é muito complexa, quando seus índices precisam ser precisos, ou quando o conhecimento necessário para compreender o caso não está disponível de forma concreta ou é muito complexo para ser representado computacionalmente. A seleção manual de índices requer o auxílio de um engenheiro de conhecimento e, é realizada através dos passos:

1. O engenheiro de conhecimento determina quais casos são úteis para representar o conhecimento do domínio e em quais circunstâncias eles poderão ser lembrados.
2. Um conjunto de descritores é selecionado a partir do conjunto de decisões de raciocínio que o caso possa suportar.
3. Os descritores são testados para assegurar que os índices escolhidos são geralmente aplicáveis, e que cada descritor foi descrito de maneira reconhecível.
4. Ao final um conjunto de descritores é determinado.



## 2.3 Análise de Domínio

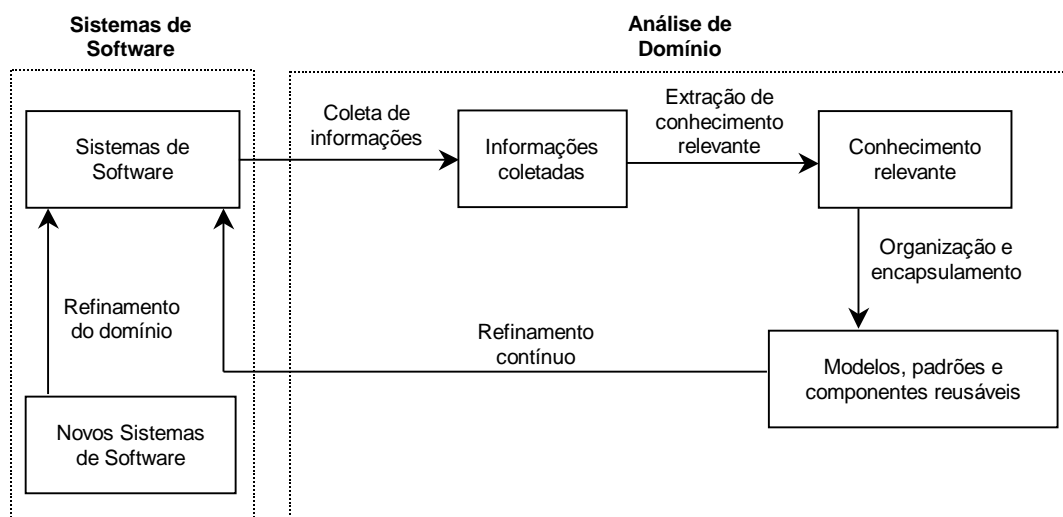
A análise de domínio é uma possível resposta para que um desenvolvedor possa, confiável e sistematicamente, capturar as informações necessárias para maximizar a reutilização no processo de desenvolvimento de software [DA91].

A análise de domínio é considerada uma área de pesquisa em evolução, até alguns anos atrás, de acordo com o “*Domain Analysis Working Group Report*” [Tra92], a análise de domínio era considerada um campo de pesquisa sobre a qual não havia sido alcançado consenso. Atualmente, porém, começa a existir uma convergência entre os diversos trabalhos existentes em análise de domínio. Prova disto é que desta convergência nasceram algumas definições aceitáveis para área e foram identificados problemas comuns a serem enfrentados durante o processo de análise, além de caminhos obrigatórios a serem seguidos pelos métodos de análise de domínio e pontos que ainda necessitam pesquisa [CW97].

A análise de domínio é definida por J. M. Neighbors [Nei81] como a forma de identificar objetos, operações e relacionamentos entre tudo o que os especialistas em um domínio julgam importante para este domínio. Segundo R. Prieto-Díaz [Dia90], análise de domínio é o processo de identificar, capturar e organizar a informação utilizada no desenvolvimento de software, com o propósito de torná-la reutilizável para novos sistemas. Já G. Arango [Ara94] define análise de domínio como a atividade de identificação, aquisição e avaliação de informação reutilizável no domínio do problema, para ser utilizada na especificação e construção do sistema de software. Apesar de algumas diferenças entre as definições citadas, pode-se perceber que basicamente a análise de domínio permite a identificação e a organização do conhecimento sobre determinado domínio de modo a torná-lo reutilizável em novas situações, geralmente novos sistemas de software.

Diversos tipos de informação são produzidos durante o desenvolvimento de software, desde documentos de requisitos até o código propriamente dito.

Ao decidir que determinado componente será reutilizável, é necessário identificar o contexto original no qual o desenvolvedor construiu o componente. O sucesso do processo de análise de domínio é traduzido na identificação correta do domínio, e na seleção e abstração de operações e objetos essenciais que irão caracterizá-lo. O produto obtido através da análise de domínio é encapsulado na forma de procedimentos, arquiteturas genéricas ou linguagem formal. De forma resumida, a análise de domínio procura identificar e estruturar o conhecimento para que ele possa ser reutilizado futuramente. A figura 3 mostra o processo de análise de domínio apoiando o processo de reutilização de software.



**Figura 3: A aplicação da técnica de análise de domínio para auxiliar o processo de reutilização de software**

Nas próximas seções, são apresentados o processo de análise de domínio e alguns dos principais métodos utilizados na área.

### 2.3.1 Processo de Análise de Domínio

O processo de análise de domínio pode ser descrito, de forma resumida, pelas seguintes etapas:

1. Coleta de informações sobre os sistemas existentes;
2. Extração de informações e conhecimento relevantes sobre os dados coletados;
3. Organização e encapsulamento em forma de modelos, padrões e coleções de componentes reutilizáveis do conhecimento e abstrações extraídos;

O processo descrito acima é continuamente refinado à medida que novos recursos e novos sistemas são construídos. O conhecimento obtido através da construção destes sistemas é avaliado e acrescentado à biblioteca de reutilização.

A tarefa do analista de domínio<sup>4</sup>, em conjunto com um ou mais especialistas no domínio<sup>5</sup>, é extrair informação relevante e conhecimento sobre o domínio e também analisar e abstrair o conhecimento coletado. Com o apoio de um engenheiro de domínio<sup>6</sup>, o conhecimento e as abstrações são organizados e encapsulados na forma de uma arquitetura de domínio. Padronizações são definidas e componentes reusáveis são criados e armazenados em uma biblioteca de reutilização.

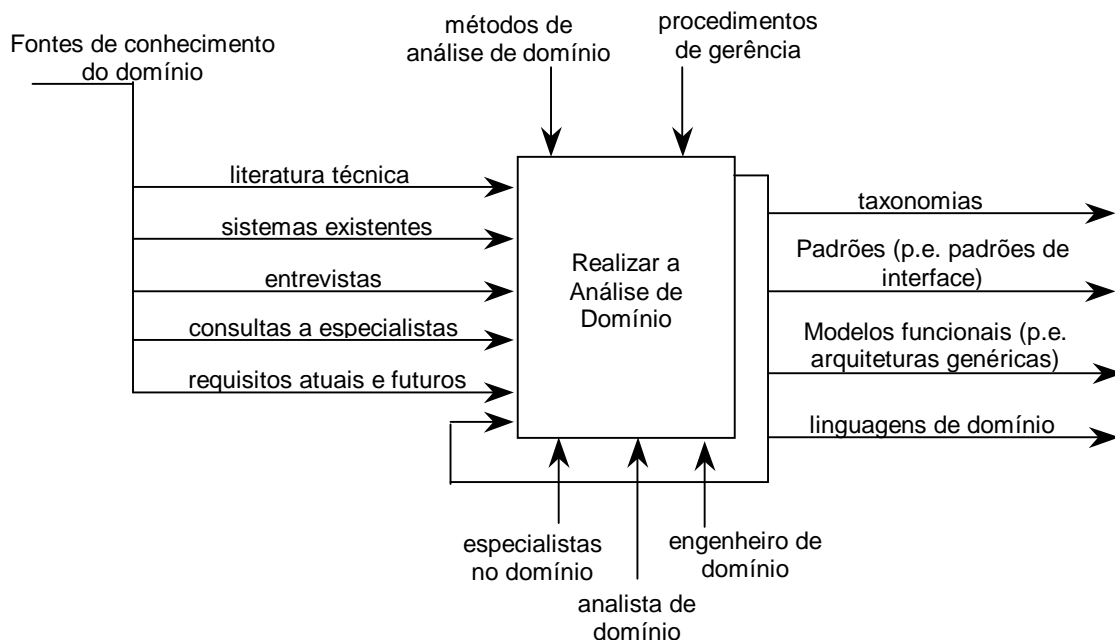
---

<sup>4</sup> O analista de domínio é o responsável pela condução do processo de análise de domínio.

<sup>5</sup> O especialista no domínio é o responsável pelo fornecimento de informações sobre o domínio.

<sup>6</sup> O engenheiro de domínio é o responsável pelo encapsulamento dos dados adquiridos durante a fase de coleta de conhecimento.

Para ilustrar melhor o processo de análise de domínio, é apresentado o diagrama de contexto SADT [Dia90] na figura 4. Neste diagrama são apresentados as entradas, saídas e mecanismos envolvidos na análise de domínio.



**Figura 4: Diagrama de contexto SADT para o processo de análise de domínio (fonte: [Dia90])**

### 2.3.2 Métodos de Análise de Domínio

Os métodos de análise de domínio têm como objetivo estruturar o processo de análise de domínio definindo as tarefas a serem executadas e os produtos a serem gerados. Muitos métodos têm sido propostos recentemente para aumentar a reusabilidade no processo de desenvolvimento de software. Segundo [DS98], grande parte dos métodos argumenta que é aplicável em qualquer tipo de domínio, porém, na prática isto nem sempre acontece. Dependendo do contexto, cada método apresenta pontos fortes e fracos. Em determinados contextos uma boa opção seria a combinação de partes de algumas abordagens de modo que o resultado pudesse ser adaptado as características do projeto de desenvolvimento de software.

Os métodos de análise de domínio possuem diferenças consideráveis com relação aos produtos gerados e passos necessários para sua execução, além disso, não existem guias para a escolha, de modo que nem sempre é possível determinar qual método melhor se aplica a determinado domínio. Apesar das diferenças, a maior parte dos métodos de análise de domínio possui quatro atividades que são comumente realizadas:

- Aquisição de conhecimento sobre o domínio;
- Análise e classificação das entidades de domínio;
- Estruturação de conhecimento do domínio;
- Construção de estruturas reutilizáveis.

Nas próximas seções alguns métodos de análise de domínio e suas características são apresentados.

#### **2.3.2.1 FODA (*Feature-Oriented Domain Analysis*)**

Desenvolvido pelo *Software Engineering Institute* (SEI) [SEI94], o propósito deste método é fornecer *background* para a exploração sistemática de sistemas de software de modo a descobrir características comuns entre eles.

O método consiste de um estudo cuidadoso do domínio (análise de contexto) que produz como resultado os produtos mostrados no quadro 5.

Passo	Produto
Análise do contexto e definição do escopo	Modelo de contexto;
Modelagem do domínio	Modelos de características ( <i>feature models</i> ); Modelos de entidade-relacionamento; Modelos de fluxo de dados e máquinas de estados finitos;
Modelagem arquitetural	Modelos de interação de processos e diagramas de estruturas.

**Quadro 5: Etapas e produtos do método FODA**

O método FODA utiliza técnicas de modelagem já bem conhecidas na Engenharia de Software, tais como as máquinas de estados e o modelo entidade-relacionamento, isto representa um grande avanço com relação a outros métodos que propõem a identificação de objetos e operações, porém não recomendam a utilização de qualquer enfoque.

O ponto fraco do método é que ele não é capaz de apoiar o relacionamento entre modelos, resultando em problemas de manutenção e evolução dos modelos.

A descrição detalhada do funcionamento do método FODA pode ser encontrada em [CW95, Ara94, SEI94].

### **2.3.2.2 ODM (*Organization Domain Modeling*)**

Esta abordagem foi desenvolvida por Mark A. Simos [ODM95] e propõe um modelo de desenvolvimento com reuso que inclui: a utilização de um conjunto de processos e métodos para realização da modelagem de domínio de uma forma sistemática; a formalização da modelagem de domínio como um ciclo de vida de transformações de modelos; a integração de aspectos técnicos da modelagem e o uso de técnicas para apontar oportunidades de reuso

baseado em geração e composição. As atividades e produtos gerados por elas são apresentados no quadro 6.

Passo	Produto
Planejamento do domínio	Modelos de “ <i>Stakeholders</i> ” Modelo de interconexão; Modelo genealógico; Exemplos de aplicações;
Modelagem do domínio	Dicionário; Modelo de conceitos; Modelos de “ <i>features</i> ”;
Refinamento do modelo	Modelo validado; Modelo interpretado; Modelo de inovações.

**Quadro 6: Etapas e produtos do método ODM**

Informações mais detalhadas sobre o método ODM podem ser encontradas em [ODM95, CW95].

### 2.3.2.3 Synthesis

Desenvolvido pelo *Software Productivity Consortium* (SPC) [SPC02], o propósito deste método é permitir a construção de sistemas de software como instâncias de uma família de sistemas que possuam características similares.

Este enfoque propõe a criação de componentes reusáveis para compor instâncias de sistemas. Devido à necessidade de cumprir o padrão DOD-STD-2167A, a análise de domínio na abordagem *Synthesis* é apresentada como uma atividade da fase de análise de requisitos do desenvolvimento de software. As etapas e produtos gerados pela utilização do método *Synthesis* são apresentados no quadro 7.

Passo	Produto
Descrição do domínio	Taxonomia conceitual e glossário de domínio;
Qualificação do domínio	Análise de riscos e negócios;
Criação de uma base de conhecimento	Base de conhecimento sobre o domínio;
Especificação de requisitos	Modelo de requisitos, decomposição dos requisitos.

**Quadro 7: Etapas e produtos do método Synthesis**

Outras informações sobre o Synthesis podem ser encontradas em [Ara94, DS98, CFW90, SPC02].

### 2.3.2.4 Método Comum de Arango

Desenvolvido por Guillermo Arango [Ara94], este método foi elaborado a partir de um levantamento das similaridades existentes entre diversas abordagens para o processo de análise de domínio. Neste levantamento, foram analisados os seguintes métodos: Prieto-Díaz [Dia87], McCain [McC85], Organon [Sim91], IDEa [Lub91], RAPID [VG90], KAPTUR [Bai91], FODA [SEI94] e Synthesis [CFW90].

O método comum de Arango é composto de cinco atividades. A primeira atividade é uma etapa de planejamento e análise de viabilidade. Nesta atividade o domínio é selecionado, delimitado e definido. Posteriormente é iniciado o processo de busca por fontes de informação sobre o domínio. A segunda atividade é a coleta de dados. As abordagens mais usadas nesta atividade são: recuperação de abstrações, pesquisa na literatura, elicitación do conhecimento (através de entrevistas, questionários, experimentações, etc.) e o desenvolvimento de cenários do domínio. A atividade três verifica a corretude, consistência e completude dos dados coletados na atividade dois. Informações irrelevantes são descartadas e dados importantes são organizados. Na quarta etapa, os dados são classificados. Descrições similares são agrupadas, classificadas e generalizadas. Um vocabulário para o domínio

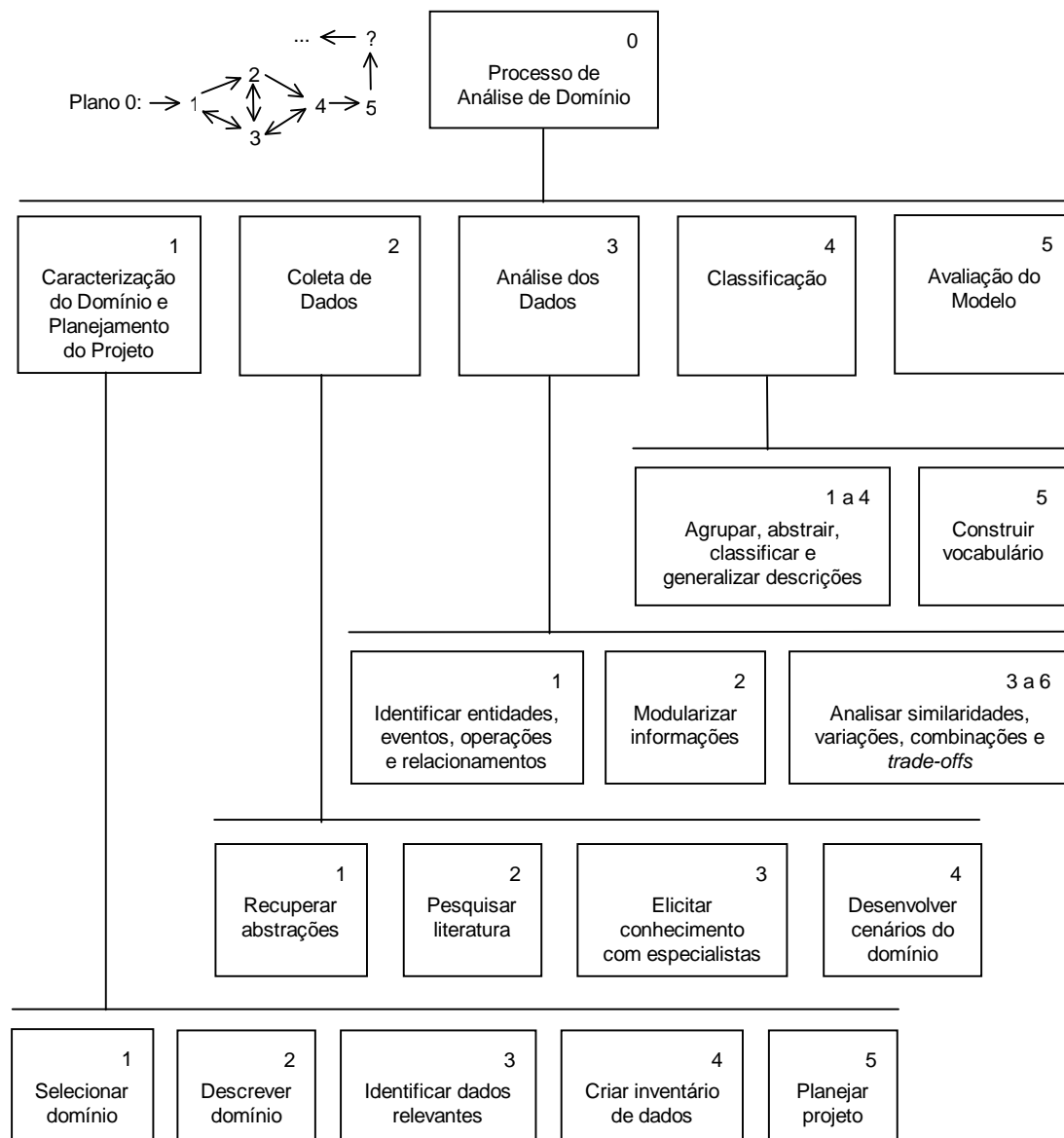


é construído. A última etapa do método consiste numa atividade de avaliação [Ara94].

Uma visão geral do método é mostrada na figura 5, utilizando a notação HTA (*Hierarchical Task Analysis*) [Dia89]. Esta notação representa uma decomposição hierárquica de atividades, segundo níveis ou planos. Os passos de cada plano são representados como caixas numeradas, enquanto que um diagrama de fluxo descreve a seqüência destes passos e o ponto de interrogação representa uma indagação sobre qual é o próximo passo a ser dado. Embora tal notação não possibilite a representação dos produtos e entradas de cada atividade, ela permite uma visão geral e simplificada do método. As principais atividades e produtos gerados pelo método comum de Arango são mostrados no quadro 8.

Passo	Produto
Caracterização do domínio	Descrição e delimitação do domínio, e planejamento do projeto;
Coleta de dados	Elaboração de cenários sobre o domínio;
Análise de dados	Definição de eventos, entidades, operações e relacionamentos existentes no domínio;
Classificação	Agrupamento de informações, definição de abstrações e generalizações, e construção de um vocabulário;
Avaliação	Avaliação do modelo.

**Quadro 8: Etapas e produtos do método comum de Arango**



**Figura 5: Diagrama do método comum de Arango (fonte: [Ara94])**

### **3. METODOLOGIA PARA INDEXAÇÃO DE EXPERIÊNCIAS DE DESENVOLVIMENTO DE SOFTWARE**

A abordagem desenvolvida neste trabalho possui o propósito de determinar um conjunto de características contextuais que sejam relevantes e discriminantes e permitam auxiliar o processo de representação, recuperação e reutilização de experiências de desenvolvimento de software. Através da aplicação da abordagem um conjunto de características é determinado para indexar uma base de experiências construída por organizações que desenvolvam sistemas de software. O método parte do princípio que o contexto, no qual as experiências se originaram, é importante para sua representação, e por isso precisa ser determinado para aumentar o aproveitamento na recuperação de casos do passado e na reutilização futura do conhecimento.

Na próxima seção, um resumo das etapas do método é apresentado, e nas seções seguintes cada passo da metodologia é mostrado em maiores detalhes.

#### **3.1 Etapas do Método**

A abordagem constitui-se de seis etapas. Na primeira etapa uma ou mais metas de reutilização são determinadas. A meta de reutilização consiste do objetivo a ser atingido com o uso da reutilização. Através da definição da meta é possível avaliar se determinadas características têm maior ou menor importância conforme o contexto abrangido pela meta.

O segundo passo do enfoque consiste em uma fase de aquisição de conhecimento sobre o domínio das experiências. Nesta etapa alguns casos são coletados e armazenados. Os casos formam uma pequena base de

conhecimento que é utilizada para dar uma idéia inicial sobre o domínio de problemas abrangido no processo de reutilização.

Na terceira etapa é aplicado um questionário para aquisição de informações sobre a organização. Neste questionário, dados como, o setor industrial que a empresa participa e a forma de estruturação do trabalho na organização, são coletados e armazenados para posterior análise em outra etapa do método.

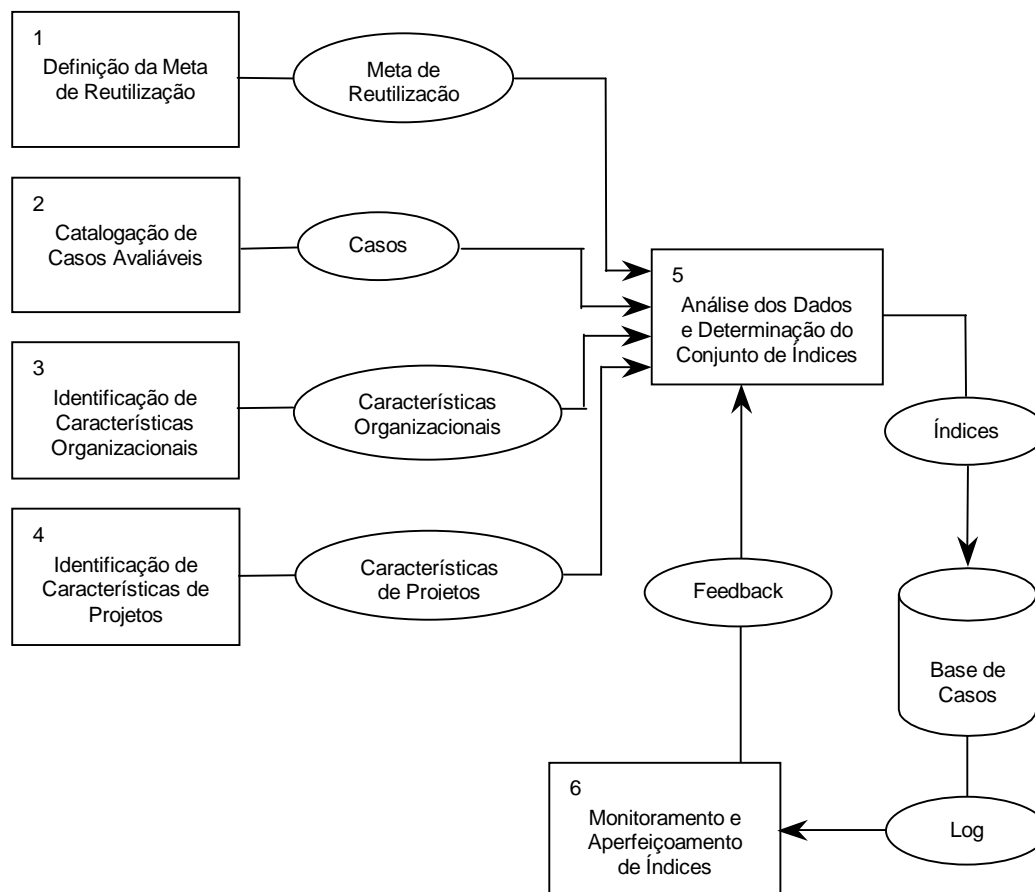
O quarto passo tem como objetivo adquirir conhecimento sobre os projetos desenvolvidos pela organização. Nesta etapa um questionário é aplicado sobre cada projeto de software da organização. Os dados obtidos dos questionários são coletados e armazenados.

No quinto passo inicia-se o processo de análise e seleção de índices. Primeiramente os questionários são analisados e um conjunto inicial de características discriminantes é automaticamente sugerido através da aplicação de alguns algoritmos para seleção de características. Cada característica deste conjunto é então avaliada e um fator de relevância é definido levando em consideração a meta de reutilização anteriormente determinada. Ao final um conjunto de índices juntamente com o peso de cada índice é proposto.

A sexta etapa é uma fase de monitoramento e avaliação do conjunto de índices determinado. A finalidade desta etapa é criar formas de verificar se os índices foram escolhidos apropriadamente, conforme a meta selecionada. Para isso, são utilizados processos de monitoramento que permitem avaliar a performance do conjunto de índices gerados. Através das informações adquiridas nesta etapa é possível verificar a boa representatividade do conjunto de índices determinado na etapa anterior. Caso a performance dos índices escolhidos não seja satisfatória então planos de reestruturação são sugeridos para modificar os índices inicialmente definidos.

A figura 6 mostra o diagrama de etapas da metodologia. Neste diagrama, os retângulos representam cada etapa do método, os círculos

contém os produtos gerados por cada passo, e as setas denotam o fluxo de dados entre as etapas.



**Figura 6: Diagrama mostrando o fluxo de informações da metodologia proposta**

### 3.1.1 Etapa 1: Identificação da Meta de Reutilização

O objetivo desta etapa é identificar e caracterizar uma ou mais metas de reutilização. A meta de reutilização corresponde ao objetivo que deve ser atingido através da utilização do conhecimento adquirido no passado em novas situações similares dentro de uma mesma organização. Cada meta é composta de um conjunto de informações que depois de identificadas apóiam a

determinação de quais características/fatores possuem maior ou menor importância no processo de reutilização.

A determinação das metas é realizada em reuniões para discussão de temas relacionados ao processo de reutilização do conhecimento adquirido pelo desenvolvimento de software na organização. Nestas reuniões devem participar as pessoas envolvidas no desenvolvimento de projetos da organização e também deve existir uma pessoa encarregada de cuidar do processo de reutilização, a qual chamamos de engenheiro de conhecimento<sup>7</sup>. As reuniões têm como principal finalidade identificar as principais necessidades, problemas e objetivos das pessoas envolvidas no processo de reutilização. Após a realização destas reuniões, o engenheiro de conhecimento analisará as informações levantadas e elaborará um documento contendo as principais metas identificadas.

As metas de reutilização irão auxiliar a identificação de características com maior ou menor relevância para o processo de reutilização. Elas seguem um padrão semelhante à definição de metas de um plano GQM<sup>8</sup> (*Goal/Question/Metric approach*) de mensuração. O quadro 9 apresenta o modelo estrutural das metas de reutilização.

---

<sup>7</sup> O engenheiro de conhecimento é a pessoa encarregada de cuidar do andamento do processo de reutilização. Ele será o responsável pelo planejamento e execução da metodologia de indexação proposta, além de monitorar os resultados obtidos pela aplicação de cada etapa do método.

<sup>8</sup> GQM é uma abordagem para mensuração orientada a metas e aplica-se principalmente a projetos de software. Maiores informações sobre a abordagem GQM podem ser encontradas em [BCR94b, WRR98, GHW95].

Dimensão	Definição	Tipos / Exemplos
Objeto	O que será / poderá ser reutilizado? (Quais tipos de experiência farão parte da base de experiências?).	Modelos de Processo Modelos de Produto Modelos de Qualidade Lições Aprendidas Dados Problemas / Soluções Material de Treinamento Manuais FAQ ( <i>Frequently Asked Questions</i> ) Outro...
Objetivo	Para que finalidade o objeto será reutilizado?	Controle Prevenção de Falhas Melhoramento Planejamento Resolução de Problemas Avaliação Caracterização Monitoramento Outro...
Ponto de Vista	Quem irá utilizar as experiências no processo de reutilização?	Gerência Administrativa Gerência de Projeto Equipe de Projeto Grupo de Garantia de Qualidade Cliente Outro...
Processo	Em qual fase do desenvolvimento as experiências serão reutilizadas?	Análise de Requisitos Projeto Implementação Manutenção Gerência de Configuração Testes Processo de Mensuração Processo de Desenvolvimento
Contexto	Em que ambiente serão reutilizadas as experiências?	Organização Departamento Área Setor Grupo Outro...

**Quadro 9: Estrutura de uma meta de reutilização**

### 3.1.2 Etapa 2: Caracterização e Estruturação de Casos Avaliáveis

O objetivo desta etapa é construção de uma pequena base de conhecimento para delimitar o domínio de problemas envolvidos no processo de reutilização. Com este intuito, dados são coletados e armazenados em forma de casos.

A construção deste conjunto de casos é feita através da catalogação de experiências passadas realizadas em projetos da organização. Se a organização já possui uma base de conhecimento em forma de casos então esta deve ser utilizada para delinear o domínio de conhecimento abrangido. Caso contrário, sugere-se a aplicação do enfoque de lembranças (discutido na seção 2.2.2) para a definição de um conjunto de experiências.

Cada caso presente na base de conhecimento é composto de quatro partes: problema, causa, solução e resultado. O quadro 10 mostra a estrutura completa de um caso.

Estrutura de um Caso		
Caso	Problema	Descrição
		Meta
		Restrições
		Características do problema
	Causa	Descrição
		Circunstâncias
	Solução	Descrição
		Passos
		Justificativas
		Soluções alternativas
		Expectativas
		Soluções não aceitáveis
	Resultado	Descrição
		Valor (sucesso ou falha)
		Explicação para falha ou violação
		Estratégia de reparo
		Precauções para evitar o problema
		Detalhes da aplicação da solução



		Restrições à solução aplicada
--	--	-------------------------------

**Quadro 10: Estrutura de um caso**

### **3.1.3 Etapa 3: Identificação de Características Organizacionais**

Esta etapa tem como finalidade identificar características que descrevam o contexto organizacional, em organizações que têm como objetivo de negócio a produção de software ou possuem áreas e/ou equipes dentro da organização que desenvolvem software como complemento ao seu produto ou serviço. Durante esta fase são coletadas informações com o intuito de identificar aspectos sobre o contexto organizacional onde são desenvolvidos os projetos de software.

A forma utilizada para aquisição das informações sobre a organização é a aplicação de um questionário contendo questões relacionadas às características organizacionais. Este questionário deverá ser preenchido por uma pessoa que conheça detalhes da organização da companhia. O questionário inclui questões sobre:

- Tipo da organização;
- Setor industrial da empresa;
- Estrutura organizacional;
- Atividades desenvolvidas;
- Área de atuação;
- Objetivo(s) de negócio.

O questionário organizacional inicialmente proposto é mostrado no anexo 6.2. Este questionário pode ser alterado conforme a necessidade da

organização. Novas questões consideradas relevantes pelo responsável pelo preenchimento do questionário poderão ser incluídas e questões consideradas de pouca valia poderão ser removidas. O produto desta etapa é composto pelas respostas dadas ao questionário. Estas respostas contêm informações importantes sobre a organização e serão analisadas posteriormente em outra etapa do método.

#### **3.1.4 Etapa 4: Coleta de Informações sobre Projetos realizados pela Organização**

O objetivo desta etapa é coletar informações que descrevam os projetos de software realizados ou em andamento pela organização. Para isso é aplicado um questionário em cada projeto com questões referentes ao projeto, características do produto desenvolvido e recursos utilizados durante o desenvolvimento de sistemas de software. Os principais aspectos considerados no questionário são:

- Quanto ao projeto:
  - Tipo;
  - Natureza;
  - Duração;
  - Custos;
  - Aspectos de Qualidade;
  - Fatores de risco;
  - Fatores de influência.
- Quanto ao processo:

- Modelo de ciclo de vida utilizado;
- Linguagem de programação;
- Metodologia de análise;
- Metodologia de testes;
- Estilo de documentação;
- Métodos de garantia de qualidade.
- Quanto ao produto:
  - Área enfocada;
  - Público alvo;
  - Principais características (simplicidade, interface intuitiva, velocidade);
  - Plataforma.

Cada questionário de projeto deverá ser preenchido por seu respectivo gerente de projeto. O produto desta fase será composto das respostas dos questionários de projeto. Estas informações são utilizadas na próxima etapa do método, onde os dados coletados são analisados e um conjunto de índices é produzido.

O questionário de projeto inicialmente proposto é apresentado no anexo 6.3. Caso alguma questão considerada relevante não conste do questionário, ela poderá ser acrescentada ao questionário. Do mesmo modo, questões consideradas sem relevância nos projetos da organização poderão ser excluídas ou deixadas sem resposta.

### **3.1.5 Etapa 5: Análise e Interpretação dos Dados coletados e Definição de um Conjunto de Índices**

Esta etapa consiste da análise dos dados coletados nas fases anteriores para a construção de um conjunto de características indexadoras da base de conhecimento da organização. Ela é dividida em três partes:

- Definição de um conjunto inicial de características relevantes;
- Identificação da relevância do conjunto de características iniciais baseado nas metas de reutilização;
- Definição do conjunto final de índices.

#### **3.1.5.1 Etapa 5.1: Definição de um Conjunto Inicial de Características Relevantes**

O objetivo da etapa 5.1 é selecionar um conjunto inicial de características que representem as experiências que comporão a base de conhecimento. Nesta etapa é necessário que o questionário organizacional (etapa 3) esteja respondido e que pelo menos dois questionários de projeto (etapa 4) tenham sido aplicados. Os dados dos questionários são comparados (segundo algoritmos que serão explicados logo a seguir nesta seção), e características contidas nestes questionários são inicialmente selecionadas.

Primeiramente, com relação ao questionário organizacional serão considerados possíveis índices as questões que tiverem como resposta mais de um item assinalado. O quadro 11 mostra o algoritmo utilizado para aquisição de características organizacionais discriminantes.

<p>SE número de itens assinalados = 1 ENTÃO</p> <p>A questão é considerada sem relevância</p> <p>SE número de itens assinalados &gt; 1 ENTÃO</p> <p>É um possível índice, nesse caso adicionar os itens assinalados na resposta como dentro da faixa de valores possíveis para o índice</p> <p>FIM SE</p>
---

**Quadro 11: Algoritmo para definição das características iniciais do questionário organizacional**

Após a análise do questionário organizacional, os questionários de projeto são comparados para verificar diferenças entre as respostas dadas. Onde existirem diferenças é sinal de que a característica pode se tornar um possível índice para base de casos. Caso os vários projetos possuam sempre o mesmo valor para determinada característica, esta característica então será descartada como possível fator de indexação. O quadro 12 apresenta o algoritmo utilizado para a escolha inicial de características relevantes entre os questionários de projetos.

Como resultado desta fase, um conjunto geral de possíveis características indexadoras da base de conhecimento é determinado. Este conjunto será refinado nas próximas etapas até a obtenção de um conjunto final de índices.

```

Considerando
R1 = Resposta da questão X do questionário 1
R2 = Resposta da questão X do questionário 2
...
Rn = Resposta da questão X do questionário n

PARA 1 ATÉ Número de questões Faça
    SE R1 = R2 = Rn ENTÃO
        A característica é considerada sem relevância

    SENÃO (alguma das respostas é diferente)
        A característica é selecionada e os valores assinalados serão
        inseridos na faixa de valores possíveis para o índice

    FIM SE
FIM PARA

```

**Quadro 12: Algoritmo para definição do conjunto inicial de características de projeto**

### **3.1.5.2 Etapa 5.2: Identificação da Relevância (*ranking*) do Conjunto de Características Iniciais**

Nesta etapa são identificados valores de relevância para o conjunto de características selecionadas na fase 5.1. A escolha dos valores de relevância associados a cada característica é baseada nas metas de reutilização obtidas na etapa 1 e no conjunto de experiências adquiridas na etapa 2. Para auxiliar a definição deste ranking, reuniões são sugeridas. Nestas reuniões, o engenheiro de conhecimento irá elaborar documentos destacando as metas de reutilização identificadas e os tipos de problemas contidos na base de conhecimento. Através da apresentação destas informações, os gerentes de projeto irão analisar as características inicialmente selecionadas e proporão um valor de relevância para cada uma delas. Os valores de relevância deverão seguir algum critério, como exemplo é sugerido o seguinte critério:

- valor **0**: característica sem relevância;
- valor **1**: característica com pouca relevância;
- valor **2**: característica relevante;
- valor **3**: característica muito relevante;
- valor **4**: característica indispensável.

O resultado desta etapa é a definição de um conjunto de características discriminantes e seus valores de relevância (*rankings*) referentes às metas de reutilização determinadas anteriormente.

### **3.1.5.3 Etapa 5.3: Definição do Conjunto Final de Índices**

A fase 5.3 objetiva construir um conjunto final de características (índices) que irão indexar a base de casos, representando assim o contexto organizacional e os projetos desenvolvidos pela organização.

Para definição dos índices, o engenheiro de conhecimento, utilizando o conjunto de características identificadas na etapa 5.1 juntamente com os seus valores de relevância levantados na etapa 5.2, irá aplicar um algoritmo, detalhado a seguir, para determinar o conjunto final de características e os seus pesos correspondentes. A definição final dos pesos para cada característica é feita através do seguinte algoritmo (o quadro 13 apresenta a fórmula utilizada pelo algoritmo):

- Primeiramente, soma-se o valor de relevância de todas as características determinado na etapa 5.2, obtendo um valor total;

- Os valores de relevância de cada característica são então divididos pelo valor total;
- O resultado obtido será um conjunto de valores entre 0 e 1, cada um destes valores é então multiplicado por 100 e o produto corresponderá ao peso da característica.

$$\text{Peso do índice} = \frac{\text{valor de relevância do índice} * 100}{\sum \text{valores de relevância dos índices}}$$

**Quadro 13: Fórmula para o cálculo do peso de cada índice**

Se o peso de uma característica for igual a 0 (zero), a característica é considerada sem relevância e será excluída do conjunto final de características. O produto desta etapa é o conjunto final de características indexadoras da base de conhecimento e o respectivo peso de cada característica.

### **3.1.6 Etapa 6: Monitoramento e Aperfeiçoamento do Conjunto de Índices**

Após a definição do conjunto final de índices, é necessário averiguar se os índices realmente estão sendo úteis na sua tarefa de representar a base de casos de desenvolvimento de software. Com o objetivo de analisar e aprimorar constantemente o conjunto de índices são propostos alguns procedimentos para monitorar a performance dos índices durante a recuperação de casos e solucionar problemas causados pela escolha incorreta de índices ou valores de relevância.

Inicialmente, sugere-se que os sistemas computacionais baseados em RBC possuam mecanismos que criem artefatos (geralmente arquivos de *log*)



para armazenar informações relevantes obtidas durante o processo de recuperação de casos. A coleta destes dados irá permitir a análise da utilidade dos índices determinados pela aplicação da metodologia proposta. Estes artefatos devem conter informações com o propósito de:

- Identificar a quantidade de acessos aos índices utilizados na recuperação. Estes dados servem para determinar os principais índices utilizados no processo de recuperação e também os índices que estão sendo pouco ou não utilizados;
- Identificar valores de atributos não relacionados anteriormente;
- Identificar novas características não previamente relacionadas como características indexadoras de casos;
- Catalogar as situações que o usuário redefiniu os valores de relevância inicialmente propostos pelo sistema;
- Catalogar as situações em que o usuário colocou o valor “desconhecido” para determinada característica durante o processo de recuperação;
- Salvar o número de vezes que o sistema reutilizou algum caso selecionado e qual caso foi reutilizado;
- Catalogar as situações em que o usuário não reutilizou nenhum caso;
- Guardar as situações em que o usuário optou por reutilizar um caso menos similar.

Um aspecto importante que deve ser monitorado diz respeito às modificações sofridas na base de conhecimento ao passar do tempo. Novos casos são adicionados, novos projetos são desenvolvidos e as características que indexam a base de experiências precisam sofrer um processo de revisão para verificar se a performance dos índices no processo de recuperação de casos continua satisfatória. Esta revisão do conjunto de índices gerados pode ser feita seguindo os seguintes critérios:

- **Periodicamente:** em períodos pré-estabelecidos o conjunto de índices é revisado (a partir dos dados coletados pelos arquivos de log) com o objetivo de verificar sua utilidade no processo de recuperação de casos;
- **Mudança de ambiente:** se alguma mudança na estrutura da organização, no processo de desenvolvimento de software ou nas metas de reutilização tiver ocorrido então se deve aplicar novamente o método de identificação de índices para representar a nova realidade e garantir um bom desempenho dos índices na recuperação de casos;
- **Problemas com o sistema:** se o sistema estiver apresentando resultados inadequados durante a recuperação de casos, então algum problema pode ter ocorrido na determinação dos índices. O método de definição de índices deverá ser aplicado novamente, e as metas de reutilização deverão ser revistas.

Um outro ponto importante que deve ser levado em consideração durante o monitoramento do conjunto de índices selecionados, refere-se a ocorrência de situações que provocam a baixa utilização do sistema de conhecimento. Estas situações devem ser previstas e ações devem ser executadas para resolver os problemas. O quadro 14 apresenta várias situações/problemas que dificultam o uso do sistema, mas podem ser resolvidos através da aplicação das ações corretivas.

Situação	Ação
Necessidade freqüente da incorporação de novas características que não foram relacionadas entre os índices.	Revisão no processo de seleção de índices.
Utilização de valores de atributos não catalogados no dicionário de dados.	Adição dos novos valores ao dicionário de dados.
Baixa utilização dos casos recuperados da base.	Possíveis problemas relacionados à similaridade entre os casos, revisar os critérios de similaridade adotados.

Uso freqüente de um conjunto restrito de índices e pouquíssima utilização dos índices restantes.	<ul style="list-style-type: none"> <li>- Revisão do critério de seleção de índices;</li> <li>- Verificar a possibilidade da exclusão de índices não utilizados.</li> </ul>
Freqüente redefinição dos valores de pesos dos índices.	Revisão no processo de determinação de índices, enfocando principalmente os valores dados aos valores de relevância.
Baixo acesso à base de casos ou baixo número de casos reutilizados.	<ul style="list-style-type: none"> <li>- O processo de reutilização está retornando casos pouco similares por isso é necessário rever o processo de seleção de índices;</li> <li>- A base de casos está com poucos casos, é necessário que novos casos sejam coletados;</li> <li>- O usuário está com dificuldades com o uso do sistema, portanto é necessário treinar o usuário para melhor utilizar o sistema.</li> </ul>
Alta utilização de casos com valores de similaridade menor que a do melhor caso (maior grau de similaridade)	Revisão nos critérios de similaridade entre casos.
Alta incidência do atributo “desconhecido” no processo de reutilização	<ul style="list-style-type: none"> <li>- Verificar se os nomes utilizados para os símbolos e dimensões de determinado índice são significativos;</li> <li>- Fornecer treinamento para os usuários do sistema sobre a terminologia utilizada para recuperação do conhecimento.</li> </ul>

**Quadro 14: Situação e ação sugerida para aperfeiçoar o conjunto de índices**

### 3.1.7 Discussão

A metodologia apresentada procura propor guias que auxiliem a determinação de índices contextuais para uma base conhecimento. O método pode ser dividido em: coleta de informações, análise dos dados coletados, definição de índices e revisão dos índices selecionados.

A fase de coleta de dados visa à aquisição de conhecimento sobre o domínio das experiências de engenharia de software produzidas pela organização. A análise dos dados é realizada com o objetivo de selecionar informações relevantes sobre este domínio, e determinar um conjunto de características que representem as experiências adquiridas na coleta de dados. E finalmente, a revisão dos índices propostos é utilizada para assegurar que o

processo de determinação de índices está realmente apoiando o processo de recuperação e reutilização de conhecimento.

Para aprimorar a definição e o funcionamento do método proposto é necessária a sua integração a sistemas de conhecimento RBC, de modo a testar as características do método e determinar pontos fortes e fracos que poderiam ser aprimorados para o melhoramento contínuo da abordagem.

## **3.2 Ferramenta**

Para facilitar o processo de seleção de índices foi construída uma ferramenta que apóia o uso da metodologia de indexação proposta. A ferramenta, desenvolvida em Smalltalk [NW98, Jon94] utilizando o ambiente de programação VisualWorks 5i.1 [PT96, Vis00], fornece suporte à determinação de metas de reutilização; construção da base de conhecimento inicial; aplicação dos questionários organizacional e de projetos; e seleção do conjunto final de índices e valores de relevância.

O acesso às etapas do método previstas na ferramenta é feito de modo seqüencial, sendo que na fase de análise e definição de índices é necessário o preenchimento dos dados contidos nas etapas anteriores. Nas próximas seções são apresentados os passos utilizados na ferramenta para obtenção de um conjunto de índices.

### **3.2.1 Passo 1: Definição da Meta de Reutilização**

No primeiro passo da ferramenta a meta de reutilização é definida. A figura 7 mostra a tela correspondente ao passo 1. Um conjunto inicial de valores é fornecido para cada dimensão da meta (objeto, objetivo, ponto de vista, contexto e processo), sendo possível ao usuário acrescentar novos

valores não previstos inicialmente através do acionamento do botão adicionar (figura 7). Depois de definida a meta de reutilização, o próximo passo é definir um conjunto inicial de casos.

A interface da ferramenta 'Metodologia de Indexação' apresenta o seguinte layout:

- Título da Janela:** Metodologia de Indexação
- Menu:** Arquivo, Passos
- Passo 1: Identificação da Meta de Reutilização**
  - Objeto:** Lições Aprendidas [Adicionar]
  - Objetivo:** Resolução de Problemas [Adicionar]
  - Ponto de Vista:** Equipe de desenvolvimento [Adicionar]
  - Contexto:** Empresa Complex Informática [Adicionar]
  - Processo:** Desenvolvimento de software educacional [Adicionar]
- Botões de Navegação:** << Anterior, Próximo >>, Sair

**Figura 7: Ferramenta - definição da meta de reutilização**

### 3.2.2 Passo 2: Construção da Base de Conhecimento

No passo dois são catalogados algumas experiências de desenvolvimento de software. Cada caso é salvo em uma base de dados e pode ser acessado através dos botões de navegação contidos na ferramenta. A figura 8 mostra a tela correspondente ao passo dois. Esta janela foi redimensionada na ilustração para mostrar todos os componentes do caso.

**Metodologia de Indexação**

Arquivo Passos

Passo 2: Base de Casos

Novo Salvar Excluir Primeiro << >> Último

Caso: 1 Total: 5

**Problema**  
 Descrição: O instalador de aplicativos instala a aplicação mas a aplicação instalada não funciona.  
 Meta: Instalar corretamente uma aplicação gerada pelo Evezip.

**Causa**  
 Descrição: Falta dos arquivos (BW5MP18S.OCX e MSSTKPRP.DLL)  
 Circunstâncias: Dois novos arquivos (para tocar MP3) foram acrescentados ao sistema (Everest runtime) mas não foram colocados na lista de arquivos a ser

**Solução**  
 Solução: Acréscimo e registro dos arquivos e geração de novo release  
 Passos: Acrescentar os arquivos faltantes; registrar arquivos; gerar um novo release do gerador de instaladores (Evezip).

**Resultado**  
 Valor: sucesso  
 Detalhes: Os arquivos foram acrescentados no instalador do Evezip e o aplicativo de geração de instaladores voltou a funcionar adequadamente.

<< Anterior Próximo >> Sair

**Figura 8: Ferramenta - construção da base de conhecimento**

### 3.2.3 Passo 3: Preenchimento do Questionário Organizacional

No passo três está o questionário organizacional. A figura 9 apresenta a janela utilizada na ferramenta para o preenchimento das informações. Esta janela possui uma área onde está definida a questão e outra área onde se encontram os vários valores de resposta disponíveis. Caso a resposta não esteja catalogada na lista de valores é possível adicionar uma nova resposta. As questões contidas nos questionário são acessadas pelos botões de

navegação contidos no alto à direita da janela, e através deles é possível visualizar as questões dos questionários. Além disso, também é possível customizar as questões inicialmente propostas no questionário organizacional, inserindo uma nova questão considerada relevante ou excluindo uma questão considerada irrelevante.

**Figura 9: Ferramenta - preenchimento do questionário organizacional**

### **3.2.4 Passo 4: Preenchimento dos Questionários de Projeto**

O passo quatro corresponde ao preenchimento dos questionários de projeto. A figura 10 mostra a janela utilizada para usuário entrar com os dados. Cada projeto deve estar relacionado a um questionário de projetos.

Inicialmente apenas um questionário é disponibilizado pelo sistema, sendo possível acrescentar novos questionários através do botão adicionar existente nesta janela. Assim como na etapa anterior, na tela do questionário de projetos é possível acrescentar novas respostas e questões não previstas inicialmente.

**Figura 10: Ferramenta - preenchimento do questionário de projeto**

### 3.2.5 Passo 5: Definição dos Valores de Relevância

No quinto passo é definido um conjunto de índices. O usuário então deve determinar os valores de relevância para cada característica deste conjunto levando em consideração a meta de reutilização determinada anteriormente. A figura 11 mostra a tela usada neste passo. Novas



características podem ser acrescentadas ao conjunto de características inicialmente propostas através do botão “Adicionar índice de projeto” ou “Adicionar índice organizacional”. A figura 12 apresenta a janela de definição do novo índice. Após a definição inicial dos valores de relevância, é possível calcular o peso das características através do botão “Calcular Pesos”. Ao final um conjunto de índices juntamente com o peso de cada característica é determinado automaticamente.

**Metodologia de Indexação**

Arquivo Passos

Passo 5: Análise, interpretação e definição do conjunto de índices

**Indexadores Organizacionais**

Característica	Ranking	Peso
Atividades de informática	3	12
Área(s) de destino	2	8
Objetivo de melhoramento	1	4

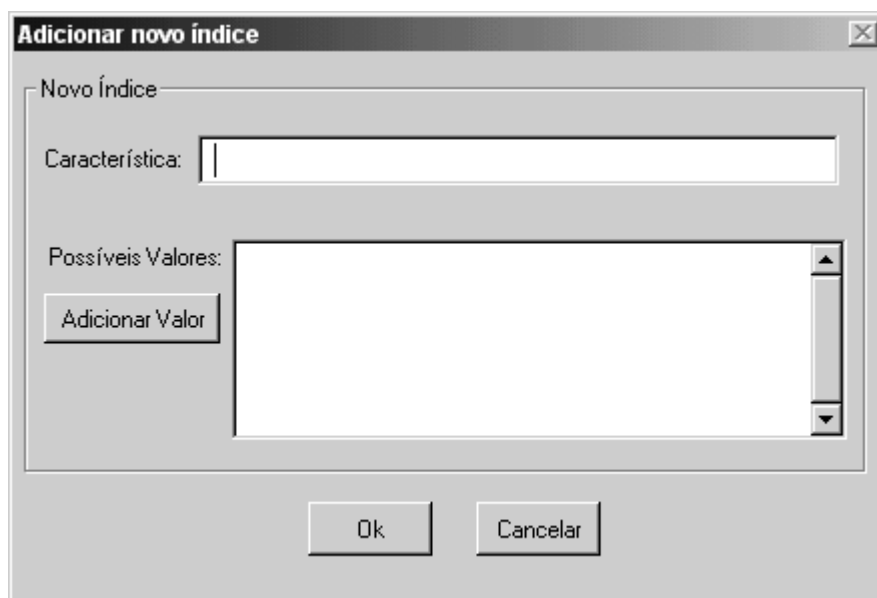
**Indexadores de Projeto**

Característica	Ranking	Peso
Tamanho da equipe	0	0
Tempo de duração	1	4
Experiência da equipe	1	4

Adicionar Índice Organizacional    Adicionar Índice de Projeto    Calcular Pesos

<< Anterior    Próximo >>    Sair

**Figura 11: Ferramenta - definição dos valores de relevância e conjunto final de índices**



The image shows a Windows-style dialog box titled "Adicionar novo índice". Inside the dialog, there is a section labeled "Novo Índice". Within this section, there is a text input field labeled "Característica:" which is currently empty. Below this, there is a list box labeled "Possíveis Valores:" which is also empty. To the left of the list box is a button labeled "Adicionar Valor". At the bottom of the dialog, there are two buttons: "Ok" and "Cancelar".

**Figura 12: Ferramenta - definição de um novo índice (passo 5)**

### **3.2.6 Passo 6: Relatório e Conjunto Final de Índices**

No último passo da ferramenta é mostrada uma tela com um relatório contendo: o conjunto final de índices, o valor de relevância de cada índice e os possíveis valores que determinada característica pode possuir. Na figura 13 é apresentada a tela contendo o relatório final.

**Metodologia de Indexação**

Arquivo Passos

**Resultados**

**Relatório Final**

**Meta de Reutilização**

Objeto: Lições Aprendidas

Objetivo: Resolução de Problemas

Ponto de Vista: Equipe de desenvolvimento

Contexto: Empresa Complex Informática

Processo: Desenvolvimento de software educacional

**Índices Organizacionais**

Característica: Atividades de informática

Possíveis valores: Desenvolvimento de software pacote

Peso: 12

<< >> Índice: 1 de 3

**Índices de Projeto**

Característica: Tempo de duração

Possíveis valores: 70

Peso: 4

<< >> Índice: 1 de 9

Imprimir relatório Salvar relatório em arquivo

<< Anterior Próximo >> Sair

**Figura 13: Ferramenta - tela mostrando o relatório final**

### 3.2.7 Discussão

A ferramenta construída procura automatizar as etapas da abordagem proposta neste trabalho. Primeiramente são coletadas informações sobre o

contexto das experiências, e posteriormente são realizadas a análise e determinação do conjunto final de índices. Apenas a última etapa do método não foi prevista pela ferramenta, porque sua implementação depende da integração da ferramenta a um sistema de conhecimento. O sistema de conhecimento disponibilizaria informações sobre o processo de recuperação de experiências para a ferramenta que utilizaria estes dados para executar o processo de verificação de performance dos índices previsto na última etapa da abordagem.

### **3.3 Aplicação e Avaliação do Método**

A metodologia de indexação proposta neste trabalho foi aplicada em uma pequena empresa do setor de desenvolvimento de software educacional, chamada Complex Informática. Este teste foi realizado com o objetivo de avaliar a aplicabilidade do método e verificar se o resultado gerado seria satisfatório. Durante a aplicação do método foram realizadas reuniões envolvendo os gerentes da organização, desenvolvedores e um engenheiro de conhecimento que ficou encarregado da aplicação da abordagem de indexação proposta. Nestas reuniões foram definidas inicialmente as metas a serem atingidas com a reutilização de conhecimento. Posteriormente, cerca de quarenta casos relacionados a problemas de suporte a usuários encontrados nos produtos desenvolvidos pela empresa foram catalogados. Os diretores da organização juntamente com a equipe de desenvolvedores ficaram encarregados do preenchimento do questionário organizacional e dos questionários de projeto.

Após o processo de aquisição de dados, as informações coletadas foram então formatadas e documentadas, e um conjunto de características foi inicialmente proposto como índices. Este conjunto de índices foi então analisado pelos diretores e pela equipe de desenvolvedores, e um conjunto

final de características foi selecionado a partir do conjunto inicial para representar o contexto organizacional e dos projetos.

Infelizmente, a organização não possuía um sistema baseado em conhecimento em operação para auxiliar no processo de reutilização de conhecimento, por isso os índices selecionados não puderam ser incorporados a uma base de conhecimento. Devido a este fato, a sexta etapa do método não foi aplicada. A documentação dos produtos gerados pela aplicação da metodologia pode ser encontrada na próxima seção.

O resultado final da aplicação do método mostrou-se bastante satisfatório em termos dos produtos obtidos e do esforço gasto para obtê-los. Durante a aplicação método ressaltaram-se alguns aspectos: primeiro, o processo de coleta de casos mostrou-se a atividade com maior esforço despendido, porque não existia um conjunto de experiências anteriormente catalogado (as experiências ficavam restritas ao conhecimento pessoal dos integrantes da equipe de desenvolvedores); segundo, a definição da meta de reutilização e a determinação dos valores de relevância mostraram-se atividades bastante importantes no processo de seleção de índices, pois permitiram um envolvimento e integração maior de todos os interessados no processo de reutilização de conhecimento.

De modo geral, a aplicação do método foi realizada de forma bastante simples sem consumir grande parcela de tempo (em menos de duas semanas foi possível definir o conjunto final de índices sendo que a maior parte do tempo foi gasto na coleta de casos) e sem dispendir grande esforço (as informações foram adquiridas de forma fácil e rápida porque as pessoas relacionadas às atividades de coleta de dados possuíam grande conhecimento sobre a organização e projetos desenvolvidos).

### 3.3.1 Teste de Aplicação do Método

O documento a seguir apresenta o resultado obtido através da aplicação da metodologia proposta na empresa Complex Informática. Os produtos de cada etapa do método são detalhados separadamente, e ao final um conjunto de índices, para representar o contexto das experiências de desenvolvimento de software, é gerado.

#### **Passo 1: Identificação da Meta de Reutilização**

A meta de reutilização selecionada pela organização foi: "resolver problemas através de lições aprendidas no desenvolvimento de sistemas educacionais do ponto de vista da equipe de desenvolvimento no contexto da empresa Complex". O quadro 15 mostra a meta escolhida.

Objeto	Objetivo	Ponto de Vista	Contexto	Processo
Lições aprendidas	Resolver problemas	Equipe de desenvolvimento	Empresa Complex	Desenvolvimento de sistemas educacionais

**Quadro 15: Teste do método - meta de reutilização selecionada pela organização**

#### **Passo 2: Caracterização e Estruturação dos Casos Avaliáveis**

Nesta etapa foram catalogados cerca de quarenta casos para auxiliarem na delimitação do contexto compreendido pelas experiências de desenvolvimento de software. Os casos compreendem um domínio de problemas bastante específico aos desenvolvedores envolvidos nos projetos da organização. Abaixo foram relacionados alguns casos dentre os vários coletados.

Caso 1	
<b>Problema</b>	1. <i>Descrição:</i> Falta de suporte no sistema ao padrão GIF e GIF animado (não é possível exibir GIFs e GIFs animadas). 2. <i>Meta:</i> fornecer o acesso ao padrão GIF e GIF animado.
<b>Causa</b>	1. <i>Descrição:</i> falta de um componente que suportasse o padrão GIF.
<b>Solução</b>	1. <i>Solução:</i> Compra e incorporação de componentes. 2. <i>Passos:</i> - Compra de novos componentes com esse padrão embutido; - Incorporação do componente ao sistema.
<b>Resultado</b>	1. <i>Resultado:</i> Sucesso (com restrições) 2. <i>Detalhes da aplicação da solução:</i> O padrão GIF e GIF animado foi implementado utilizando-se para isso um novo componente; 3. <i>Restrições:</i> - O novo controle possui problemas com GIF animados que possuam transparência (a imagem é mostrada com flicks); - Vários GIFs animados na mesma tela introduzem problemas de processamento devido à troca constante de frames.

Caso 2	
<b>Problema</b>	1. <i>Descrição:</i> O usuário não consegue executar o aplicativo aparecendo sempre ao iniciar a mensagem: "Cópia não autorizada". 2. <i>Meta:</i> detectar se a proteção está danificada.
<b>Causa</b>	1. <i>Descrição:</i> perda da proteção. 2. <i>Circunstâncias:</i> O usuário perdeu a proteção física contida no seu HD porque passou um utilitário de defragmentação de disco ou formatou o HD.
<b>Solução</b>	1. <i>Solução:</i> solicitar novo código de reset. 2. <i>Passos:</i> - Ligar para a empresa e solicitar um novo código de habilitação do programa; - Habilitar o programa através do utilitário config.exe.
<b>Resultado</b>	1. <i>Resultado:</i> Sucesso. 2. <i>Detalhes da aplicação da solução:</i> proteção restabelecida e retorno da funcionalidade original do programa; 3. <i>Solução alternativa:</i> O problema pode persistir sendo então necessário o envio de novos discos com a proteção.

Caso 3	
<b>Problema</b>	1. <i>Descrição:</i> Problemas quanto ao uso do objeto imagem de consulta com a opção pintura de polígonos. 2. <i>Meta:</i> detectar o que está provocando o problema de uso da imagem de consulta.
<b>Causa</b>	1. <i>Descrição:</i> falta de ajuda mais abrangente, acesso complexo ao recurso. 2. <i>Circunstâncias:</i> devido à falta de informações mais detalhadas no manual do sistema sobre a opção de pintura existente na imagem de consulta o usuário não consegue utilizar adequadamente este recurso, além disso, a cor de pintura da imagem de consulta está armazenada em outro objeto (placar), dificultando o

	acesso a esta propriedade para usuários iniciantes.
<b>Solução</b>	<p>1. <i>Solução</i>: melhorar o sistema de ajuda e a forma de acesso ao recurso.</p> <p>2. <i>Necessidades</i>:</p> <ul style="list-style-type: none"> <li>- Modificar a documentação do sistema com relação à pintura de imagens de consulta;</li> <li>- Criar uma nova forma de acesso à cor selecionada.</li> </ul>
<b>Resultado</b>	<p>1. <i>Resultado</i>: <b>Sucesso (com restrições)</b>.</p> <p>2. <i>Detalhes da aplicação da solução</i>: através da criação de uma ação de macro de nome: "Imagem de Consulta / Escolhe cor de pintura" ficou mais simples a utilização do recurso por parte do usuário (tanto iniciante quanto avançado).</p> <p>3. <i>Restrições</i>: o sistema de ajuda não foi totalmente concluído.</p>

<b>Caso 4</b>	
<b>Problema</b>	<p>1. <i>Descrição</i>: <b>Falta de suporte ao padrão de som MP3.</b></p> <p>2. <i>Meta</i>: implementar o padrão MP3.</p>
<b>Causa</b>	1. <i>Descrição</i> : falta de um componente que disponibilizasse o recurso de tocar arquivos MP3 na placa de som, falta de ajuda mais abrangente e acesso complexo ao recurso.
<b>Solução</b>	1. <i>Solução</i> : pesquisa e compra de um componente que possua este recurso incorporado.
<b>Resultado</b>	<p>1. <i>Resultado</i>: <b>Sucesso (com restrições)</b>.</p> <p>2. <i>Detalhes da aplicação da solução</i>: foram encontrados dois produtos com essa funcionalidade e optou-se pelo produto "X" por possuir um preço mais acessível.</p> <p>3. <i>Restrições</i>: limitação imposta pelo componente utilizado: não é possível instanciar dois novos objetos do tipo MP3 em um mesmo projeto. (Esta limitação está sendo solucionada conforme contato realizado com o fornecedor do produto).</p>

<b>Caso 5</b>	
<b>Problema</b>	<p>1. <i>Descrição</i>: <b>Durante a execução do aplicativo acontece o erro "429 - Can not create ActiveX control" e o programa é fechado.</b></p> <p>2. <i>Meta</i>: detectar falha durante a execução do sistema.</p>
<b>Causa</b>	<p>1. <i>Descrição</i>: registro incorreto de DLLs/OCXs.</p> <p>2. <i>Circunstâncias</i>: durante a instalação do sistema o banco de registros não foi corretamente gravado.</p>
<b>Solução</b>	<p>1. <i>Solução</i>: Fazer registro manual ou repetir a instalação.</p> <p>2. <i>Passos</i>:</p> <ul style="list-style-type: none"> <li>- Registrar manualmente ou através de um arquivo (.BAT) todas as DLLs e OCXs usados pelo sistema;</li> <li>- Verificar o funcionamento do sistema, caso o problema persista então repetir o processo de instalação.</li> </ul>
<b>Resultado</b>	<p>1. <i>Resultado</i>: <b>Sucesso (com restrições)</b>.</p> <p>2. <i>Restrições</i>: Em muitos casos o aplicativo volta a funcionar normalmente, alguns casos isto não ocorre sendo necessário enviar uma nova mídia de CD para o usuário.</p>



### **Passo 3: Identificação de características organizacionais**

As respostas obtidas pelo preenchimento do questionário organizacional estão expostas no quadro 16. Estas informações serão analisadas posteriormente através da aplicação do passo 5.

<b>Respostas do Questionário Organizacional</b>	
<b>Característica</b>	<b>Valores</b>
<b>Setor Industrial</b>	Educacional
<b>Atividade(s) característica da empresa</b>	Desenvolvimento de software pacote Prestação de serviços em escolas Consultoria e projetos em multimídia
<b>Tipo de software desenvolvido</b>	Software pacote
<b>Área de destino</b>	Educacional Multimídia
<b>Objetivo de melhoramento</b>	Melhorar o processo de desenvolvimento Aumentar a confiabilidade
<b>Tamanho</b>	9
<b>Estrutura de equipes</b>	Pequenas equipes (menos de 4 pessoas)
<b>Objetivo de negócio</b>	Desenvolvimento de software

**Quadro 16: Teste do método – respostas do questionário organizacional**

### **Passo 4: Coleta de informações dos projetos da empresa**

As respostas dadas aos questionários de projetos estão no quadro 17. Estes dados serão analisados na próxima etapa do método.

Respostas dos Questionários de Projetos			
Características	Projeto 1	Projeto 2	Projeto 3
Tamanho da equipe	3 pessoas	1 pessoa	1 pessoa
Duração	36 – 96 meses	12-48 meses	12-36 meses
Experiência	Média	Baixa	Baixa
Equipamento alvo	PC	PC	PC
Equipamento de desenvolvimento	PC	PC	PC
Áreas computacionais relacionadas	Multimídia	Compactação de dados Instalação	Internet
Natureza	Melhoramento	Melhoramento	Melhoramento
Classe	Externo	Interno	Interno
Tipo	Programa interativo	Programa interativo	Programa interativo
Aspectos críticos de qualidade	Funcionalidade Usabilidade	Funcionalidade Usabilidade	Funcionalidade
Fatores de influência	Facilidade de uso	Facilidade de uso	Facilidade de uso
Complexidade	Média-baixa	Média-baixa	Média-baixa
Ferramentas	- Ambiente de desenvolvimento - Ferramenta para manipulação de dados multimídia - Protetor contra pirataria	- Ambiente de desenvolvimento - Compactador de dados	- Ambiente de desenvolvimento - Conversor de imagens BMP para GIF
Modelo de ciclo de vida	Melhoramento interativo	Melhoramento interativo	Melhoramento Interativo
Metodologia de desenvolvimento	Estruturada	Estruturada	Estruturada
Métodos de determinação de requisitos	Método informal	Método informal	Método informal
Método de análise e projeto	Análise informal	Análise informal	Análise informal
Linguagem de programação	VB	VB	VB
Métodos de teste e detecção de defeitos	- Testes funcionais	- Testes funcionais	- Testes funcionais
Tipos de documentação	- Documentação do programa - Documentação da estrutura de dados - Documentação das entradas de dados - Manuais	- Documentação do programa - Documentação da estrutura de dados - Documentação das entradas de dados	Documentação do programa
Grupo de qualidade	Não	Não	Não

Quadro 17: Teste do método – respostas dos questionários de projetos

**Passo 5: Análise dos dados coletados nas fases anteriores e definição de um conjunto de índices para base de experiências**

Através da análise comparativa das respostas obtidas através da aplicação dos questionários foi produzido o conjunto inicial de índices apresentado no quadro 18.

Características iniciais		
Área	Característica	Possíveis valores
Organização	Atividade característica	Desenvolvimento de software Prestação de serviços de informática em escolas Consultoria e projetos em multimídia
	Área de destino	Educacional Multimídia
	Objetivo de melhoramento	Melhorar o processo de desenvolvimento Aumentar a confiabilidade
Projeto	Tamanho da equipe	1 – 3 pessoas
	Duração	12 – 96 meses
	Experiência	Baixa Média
	Áreas computacionais relacionadas	Multimídia Compactação de dados Internet Instalação
	Classe	Externo Interno
	Ferramentas	Ambiente de desenvolvimento Ferramenta para manipulação de dados multimídia Ferramenta para MP3 Protetor contra pirataria Compactador de dados Conversor de imagens BMP para GIF
	Tipos de documentação	Documentação de programa Documentação da estrutura de dados Documentação das entradas e saídas Manuais

**Quadro 18: Teste do método – análise comparativa dos questionários**

Além do conjunto inicial de características de contexto, foi levantada outra característica considerada relevante apresentada no quadro 19.

Outras características de contexto levantadas		
Área	Característica	Possíveis valores
Projeto	Versão do produto	1.0, 2.0, 3.0, 3.1, 3.2, 3.3, 4.0, 5.0

**Quadro 19: Teste do método – nova característica levantada**

No quadro 20 são identificados os valores de relevância para cada característica conforme a meta de reutilização determinada na fase 1.

Características de contexto e valores de relevância		
Área	Característica	Valor de Relevância
Organização	Atividade característica	3
	Área de destino	2
	Objetivo de melhoramento	1
Projeto	Tamanho da equipe	0
	Duração	1
	Experiência	1
	Áreas computacionais relacionadas	4
	Classe	3
	Ferramentas	3
	Tipos de documentação	0
	Versão do produto	3

**Quadro 20: Teste do método – identificação dos valores de relevância**

O quadro 21 apresenta o conjunto final de características e os pesos de cada índice.

Conjunto final de características		
Área	Característica	Peso
Organização	Atividade característica	14,29
	Área de destino	9,52
	Objetivo de melhoramento	4,76
Projeto	Duração	4,76
	Experiência	4,76
	Áreas computacionais relacionadas	19,05
	Classe	14,29
	Ferramentas	14,29
	Versão do software	14,29

**Quadro 21: Teste do método – conjunto final de características**

## 4. CONCLUSÃO

Neste trabalho de pesquisa foi desenvolvida uma metodologia para caracterizar o contexto do conhecimento formado por um conjunto de experiências obtidas através do desenvolvimento de projetos de software. A abordagem proposta fornece mecanismos para identificação de índices que irão representar toda base de conhecimento.

Durante o processo de elaboração da abordagem, algumas técnicas foram analisadas para dar suporte à construção do método. A técnica de Raciocínio Baseado em Casos foi estudada por tratar-se de uma abordagem adequada à construção de sistemas baseados em conhecimento e também por fornecer suporte a operacionalização de fábricas de experiências. As principais definições relacionadas à estrutura de representação de casos em RBC foram assimiladas neste trabalho. Além disso, os métodos e fundamentos utilizados na indexação de casos foram avaliados, sendo que muitos dos seus conceitos foram aplicados na definição da metodologia proposta.

Com base nos requisitos identificados, algumas abordagens de análise de domínio foram avaliadas. Muitos dos recursos utilizados, pelos métodos de análise de domínio, para aquisição de conhecimento sobre um domínio, como a aplicação de questionários, a realização de entrevistas e reuniões foram assimilados pela metodologia proposta. Além disso, alguns fundamentos de análise de domínio estão presentes na metodologia como a coleta e análise de dados, e a identificação de dados relevantes sobre determinado domínio.

Neste trabalho, uma ferramenta computacional foi construída para auxiliar a utilização da metodologia proposta. Esta ferramenta teve como principal finalidade a automatização do processo de seleção de índices de modo a tornar a tarefa de definição do conjunto de características que irão indexar a base de conhecimento mais fácil e rápida. Durante o desenvolvimento do trabalho, constatou-se que para a utilização da ferramenta ser feita de modo mais amplo e eficiente seria necessário que ela fosse integrada a um sistema computacional baseado em conhecimento que

utilizasse a técnica RBC. A integração da ferramenta a um sistema RBC apoiaria a identificação de índices contextuais e também auxiliaria o sistema no processo de monitoramento dos índices selecionados. Como sugestão de trabalho futuro, propõe-se a integração da ferramenta desenvolvida ao sistema REMEX<sup>9</sup>. A incorporação da metodologia ao sistema REMEX permitiria que os casos constantes na base de conhecimento fossem indexados tanto por suas características estruturais quanto por suas características contextuais.

A metodologia proposta neste trabalho foi avaliada de maneira inicial em um campo de teste bastante restrito, como mostrado na seção 3.4. Dentro destas circunstâncias, verificou-se que a aplicação do método foi realizada de forma rápida e simples, isto aconteceu devido principalmente à cooperação dos envolvidos no processo. Convém ressaltar também que os produtos gerados pelo método atenderam às expectativas dos interessados. Apesar dos resultados obtidos, é necessário destacar que o método deveria ser aplicado em novos contextos para averiguar se o seu desempenho é satisfatório. Além disso, seria muito importante que a metodologia fosse aplicada e incorporada a sistemas baseados em conhecimento que utilizassem a técnica de RBC para avaliar seu funcionamento integrado a este tipo de sistema e também verificar se os procedimentos de monitoramento permitem o aprimoramento dos processos de recuperação e reutilização de conhecimento.

Outra sugestão proposta seria a aplicação da abordagem para indexar bases de conhecimento compostas exclusivamente por experiências de áreas específicas de determinado processo de desenvolvimento de software, tais como a mensuração, a análise de requisitos e a implementação de código.

---

<sup>9</sup> REMEX é uma ferramenta que faz uso de Raciocínio Baseado em Casos para fornecer suporte ao planejamento do processo de mensuração de software com base na reutilização de produtos de mensuração e experiências na forma problema/solução coletados em programas de mensuração passados [Rod00, Cor01, WRW<sup>+</sup>99].

## 5. FONTES BIBLIOGRÁFICAS

- [ABT98] K.-D. Althoff, F. Bomarius, C. Tautz. **Using Case-Based Reasoning Technology to Built Software Organizations**. In Proceedings of the Workshop on Organizational Memories at the European Conference on Artificial Intelligence, 1998.
- [AP94] A. Aamodt, E. Plaza. **Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches**, AI Communications, vol. 17, nr. 1, 1994.
- [Ara94] G. Arango. **Domain Analysis Methods**. Software Reusability, pg. 17-49, 1994.
- [Bai91] S. Bailin. **Kaptur: A Tool for the Preservation and Use of Engineering Legacy**. CTA Incorporated, 1991.
- [BCR94a] V. R. Basili, G. Caldiera, H. D. Rombach. **The Experience Factory**. In John J. Marciniak editor, Encyclopedia of Software Engineering, volume 1, pg. 528-532. John Wiley & Sons, 1994.
- [BCR94b] V. R. Basili, G. Caldiera, H. D. Rombach. **The Goal/Question/Metric Paradigm**. In John J. Marciniak editor, Encyclopedia of Software Engineering, volume 1. John Wiley & Sons, 1994.
- [CFW90] G. Campbell, S Faulk, and D. Weiss. **Introduction to Synthesis**. Technical Report Intro\_Synthesis-90019-N, V 01.00.01, Software Productivity Consortium, 1990.
- [Cor01] L. S. Côrrea. **Planejamento, Execução e Análise de um Experimento para Avaliar uma Tecnologia de Mensuração de Software**. Universidade Federal de Santa Catarina, 2001.
- [CW95] A .M. Cima, C. M. L. Werner. **O Processo de Escolha de um Método de Análise de Domínio**. Programa de Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro,

- 1995.
- [CW97] A .M. Cima, C. M. L. Werner. **A Reutilização de Conhecimento Abstrato e Análise de Domínio**. Programa de Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro, 1997.
- [DA91] R. Prieto-Díaz, G. Arango. **Domain Analysis Concepts and Research Directions**. Domain Analysis and Software System Modeling, IEEE Computer Society Press Tutorial, 1991.
- [Dia87] R. Prieto-Díaz. **Domain Analysis for Reusability**. The 11<sup>th</sup> Annual International Computer Software & Application Conference, 1987.
- [Dia89] W. B. Diaper. **Task Analysis for Human-Computer Interaction**. John Wiley and Sons, 1989.
- [Dia90] R. Prieto-Díaz. **Domain Analysis: An Introduction**. ACM SIGSOFT Software Engineering Notes, 1990.
- [DS98] J. Debaud, K. Schmid. **A Practical Comparison of Major Domain Analysis Approaches – Towards a Customizable Domain Analysis**. In Proceedings of the 10<sup>th</sup> International Conference on Software Engineering & Knowledge Engineering, 1998.
- [GHW95] C. Gresse, B. Hoisl, J. Wüst. **A Process Model for GQM-Based Measurement**. Report STTI-95-04-E, Software Technology Transfer Initiative, University of Kaiserslautern, Germany, 1995.
- [Hen95] S. Henninger. **Developing Domain Knowledge Through The Reuse of Project Experiences**. 1995.
- [Hof97] M. Hoffmann. **A Decision Support System for Selection of Software Engineering Technologies Based on Models of their Application Domains**, Fraunhofer IESE, Germany, 1997.
- [Jon94] M. Jonathan. **Introdução à programação orientada a objetos com Smalltalk**. Universidade Federal do Rio de Janeiro, 1994.
- [Kol93] J. L. Kolodner. **Case Based Reasoning**, pp. 193-218, 247-281,



- 369-389, Morgan Kaufmann, 1993.
- [Lub91] M. D. Lubars. **Domain Analysis and Domain Engineering in IDea**. IEEE Computer Society Press, 1991.
- [Mcc85] R. McCain. **Reusable Software Component Construction: A Product Oriented Paradigm**. Proceedings of the 5<sup>th</sup> AIAA/ACM/NASA/IEEE Computer in Aerospace, 1985.
- [MMM95] H. Mili, F. Mili, A. Mili. **Reusing Software: Issues and Research Directions**. IEEE Transactions on Software Engineering, vol. 21, n. 6, 1995.
- [Nei81] J. M. Neighbors. **Software Construction Using Components**. University of California, 1981.
- [NW98] V. R. Nunes, A. Wangenheim. **Curso de Smalltalk usando Visualworks – módulo básico**. Universidade Federal de Santa Catarina, 1998.
- [ODM95] **Organization Domain Modeling (ODM) Guidebook version 1.0**. Prepared by Unisys Corporation, 1995.
- [PT96] A. Painter, A. J. Turner. **Introduction to VisualWorks: An Application Approach**. Clemson University Department of Computer Science. 1996.
- [Rod00] M. R. Rodrigues. **Desenvolvimento e Implementação de um Protótipo de Ferramenta para a Reutilização de Planos de Mensuração Utilizando Raciocínio Baseada em Casos**, Universidade Federal de Santa Catarina, 2000.
- [SEI94] Software Engineering Institute. **Feature-Oriented Domain Analysis (FODA)**. Domain Modeling Tutorial, Workbook, 1994.
- [Sim91] M. A. Simos. **The Domain-Oriented Software Life-Cycle: Towards an Extend Process Model for Reusability**. Workshop on Software Reusability and Maintainability, 1987.
- [SPC02] Software Productivity Consortium. **Digital Library**. Disponível na

internet. <http://www.software.org/Library/>, 2002.

- [Tra92] W. Tracz. **Domain Analysis Working Group Report – First International Workshop on Software Reusability**. ACM SIGSOFT Software Engineering Notes, 1992.
- [VG90] W. Vitaletti, E. Guerrieri. **Domain Analysis within the ISEC Rapid Center**. Proceedings of 8<sup>th</sup> Annual National Conference on ADA Technology, 1990.
- [Vis00] **VisualWorks Application Developer's Guide**. 2000.
- [Wan00] C. G. Wangenheim, **Reutilização Baseada em Casos de Experiência na Área de Mensuração em Engenharia de Software**, Universidade Federal de Santa Catarina, 2000.
- [Wan98] C. G. Wangenheim. **Knowledge Management in Experimental Software Engineering – Create, Renew, Build, and Organize Knowledge Assets**. In Proceedings of the 10<sup>th</sup> Int. Conference on Software Engineering and Knowledge Engineering, 1998.
- [Wat97] I. Watson. **Applying Case-Based Reasoning: Techniques for Enterprise Systems**, Morgan Kaufmann Publisher, 1997.
- [WRR98] C. G. Wangenheim, H. D. Rombach, G. Ruhe. **Tutorial de Melhoramento de Software Baseado em Mensuração – Como Aplicar GQM na Prática?**. IX CITS – Conferência Internacional de Tecnologia de Software: Qualidade de Software, 1998.
- [WRW<sup>+</sup>99] C. G. Wangenheim, M. R. Rodrigues, A. Wangenheim, R. M. Barcia. **O Uso de Fábricas de Experiência em Software Engineering**, Developers' Magazine, novembro, 1999.
- [WWB98] C. G. Wangenheim, A. Wangenheim, R. M. Barcia. **Case-Based Reuse of Software Engineering Measurement Plans**. In Proceedings of the 10th Int. Conference on Software Engineering and Knowledge Engineering, 1998.

## 6. ANEXOS

### 6.1 Aplicação Experimental da Análise do Domínio para Caracterizar o Desenvolvimento de Software

Este documento apresenta os produtos gerados pela aplicação do método de análise de domínio de Guillermo Arango para delimitar o domínio específico do desenvolvimento de software. O método foi aplicado com o objetivo de aumentar o conhecimento sobre o domínio enfocado e também para conhecer melhor o funcionamento de um método de análise de domínio.

Nas seções seguintes encontram-se os produtos obtidos em cada etapa do método aplicado.

#### 1. Caracterização do Domínio

**1.1. Seleção do domínio:** o domínio selecionado foi o desenvolvimento de software.

#### **1.2. Descrição do domínio:**

- Técnica utilizada: pesquisa na literatura e conhecimento de especialistas.
- Descrição: Um software é desenvolvido por uma organização com determinada finalidade e funcionalidade. O desenvolvimento de software é feito através de um projeto. Cada projeto de software é composto de etapas. Estas etapas compõem o processo de desenvolvimento de software que é, geralmente, dividido nas seguintes fases: análise de requisitos; projeto e modelagem do sistema; implementação; testes; manutenção; e mensuração. Cada projeto possui características específicas dependendo de suas finalidades e da forma que ele é desenvolvido. Todo projeto possui um conjunto de pessoas trabalhando dentro dele.

#### **1.3. Identificação de dados relevantes**

- Técnica utilizada: conhecimentos de especialistas e análise de questionários.
- Pontos em destaque:
  - Uma organização que produz software participa de um setor industrial ou comercial;
  - Os projetos de software desenvolvidos podem ou não aplicar todas as fases do processo de desenvolvimento de software descritas no domínio;
  - Muitos projetos utilizam ferramentas automatizadas para ao auxiliar o seu desenvolvimento;

- A duração do projeto é variável conforme sua complexidade e, a experiência e o tamanho de sua equipe;
- Os projetos são desenvolvidos com determinada finalidade e para determinada área;
- Os projetos possuem fatores de riscos que devem ser considerados antes e durante a sua implementação;
- Em alguns projetos são aplicados métodos de garantia de qualidade;
- Existem vários métodos de análise e projeto para o desenvolvimento de sistemas;
- Existem várias linguagens de codificação para a implementação de programas.

#### **1.4. Inventário de Dados:**

As fontes utilizadas para a aquisição do conhecimento sobre o domínio foram:

- Literatura (livros de engenharia de software);
- Análise de pesquisas e questionários publicados em revistas (publicação desenvolvida pelo ministério de ciência e tecnologia sobre o tema de qualidade no setor de software brasileiro);
- Questionários do projeto CEMP (*Customized Establishment of Measurement Programs*);
- *Screenshots* do sistema computacional Checkpoint, que têm como objetivo a determinação de custos durante o desenvolvimento de projetos de software;
- Entrevistas realizadas com gerentes do projeto GENESS da Universidade Federal de Santa Catarina.

## **2. Coleta de Dados**

**2.1. Catalogação de dados:** questões dos questionários CEMP e do questionário sobre qualidade de software foram catalogadas. Além disso, as características utilizadas como entrada para o sistema Checkpoint foram registradas.

**2.2. Entrevistas de validação com especialistas:** foram realizadas reuniões quinzenais com especialistas do grupo GENESS, durante dois meses, para discussão de temas relacionados às informações coletadas nas etapas anteriores.

## **3. Análise dos Dados**

**3.1. Modelagem e análise dos dados:** através da análise das informações obtidas nas reuniões com especialistas do grupo GENESS foram elaborados os questionários relacionados ao desenvolvimento de software.

#### **4. Classificação**

**4.1. Agrupamento de características:** os questionários elaborados foram divididos em duas partes: um questionário organizacional e outro referente aos projetos de software desenvolvidos pela organização.

**4.2. Vocabulário:** para cada questão (dimensão) do questionário algumas opções foram catalogadas (símbolos). As questões e opções constantes no questionário representam o vocabulário compreendido neste domínio.

## 6.2 Questionário Organizacional

### 1. Quais setores industriais a organização participa?

- |  |   |
|--|---|
| <input type="checkbox"/> Energia                         | <input type="checkbox"/> Turismo                  |
| <input type="checkbox"/> Mineração                       | <input type="checkbox"/> Telecomunicações         |
| <input type="checkbox"/> Manufatura de artigos metálicos | <input type="checkbox"/> Tecnologia da Informação |
| <input type="checkbox"/> Automação de processos          | <input type="checkbox"/> Educacional              |
| <input type="checkbox"/> Engenharia                      | <input type="checkbox"/> Científico               |
| <input type="checkbox"/> Comércio / Negócios             | <input type="checkbox"/> Internet                 |
| <input type="checkbox"/> Outro(s) _____                  |   |

### 2. Quais são as atividades de informática características da empresa?

- ☐ Comercialização de dados ou de base de dados
- ☐ Consultoria e projetos em informática
- ☐ Desenvolvimento de software pacote
- ☐ Desenvolvimento de software sob encomenda
- ☐ Serviços de automação (bancária, comercial, industrial)
- ☐ Serviços de coleta de dados
- ☐ Serviços de processamento de dados
- ☐ Treinamento em informática
- ☐ Outra(s) \_\_\_\_\_

### 3. Quais são os tipos de software desenvolvidos pela organização?

- ☐ Software de uso próprio
- ☐ Software pacote
- ☐ Software elaborado sob encomenda
- ☐ Software embutido ou de tempo real
- ☐ Outro(s) \_\_\_\_\_

**4. Para quais áreas os produtos/serviços desenvolvidos pela organização se destinam?**

- |  |   |
|--|---|
| <input type="checkbox"/> Administração                             | <input type="checkbox"/> Financeiro                 |
| <input type="checkbox"/> Administração de recursos humanos         | <input type="checkbox"/> Gerenciador de informações |
| <input type="checkbox"/> Administração escolar                     | <input type="checkbox"/> Gerenciador de redes       |
| <input type="checkbox"/> Administração jurídica                    | <input type="checkbox"/> Inteligência artificial    |
| <input type="checkbox"/> Agropecuária                              | <input type="checkbox"/> Meio Ambiente              |
| <input type="checkbox"/> Automação comercial                       | <input type="checkbox"/> Multimídia                 |
| <input type="checkbox"/> Automação de escritórios                  | <input type="checkbox"/> Planilha eletrônica        |
| <input type="checkbox"/> Automação predial                         | <input type="checkbox"/> Processador de imagens     |
| <input type="checkbox"/> Banco de Dados                            | <input type="checkbox"/> Processador de textos      |
| <input type="checkbox"/> Comunicação de dados                      | <input type="checkbox"/> Saúde                      |
| <input type="checkbox"/> Segurança e proteção de dados             | <input type="checkbox"/> Construção civil           |
| <input type="checkbox"/> Contabilidade                             | <input type="checkbox"/> Serviços de mensagens      |
| <input type="checkbox"/> Controle de qualidade e processos         | <input type="checkbox"/> Simulação e modelagem      |
| <input type="checkbox"/> Educacional                               | <input type="checkbox"/> Software gráfico           |
| <input type="checkbox"/> Engenharia e arquitetura                  | <input type="checkbox"/> Teleinformática            |
| <input type="checkbox"/> Entretenimento                            | <input type="checkbox"/> Transportes                |
| <input type="checkbox"/> Ferramenta de desenvolvimento de sistemas |   |
| <input type="checkbox"/> Outro(s) _____                            |   |

**5. Quais são os objetivos de melhoramento (escolher apenas os mais importantes) no desenvolvimento dos projetos de software?**

- ☐ Melhorar o processo de desenvolvimento de software  
☐ Controlar o processo de desenvolvimento de software  
☐ Diminuir custos  
☐ Aumentar a produtividade  
☐ Aumentar a confiabilidade  
☐ Aumentar a usabilidade  
☐ Outro(s) \_\_\_\_\_

**6. Qual o tamanho da organização (número de pessoas)? \_\_\_\_\_**

**7. Como estão estruturados os grupos de trabalho na organização?**

- ☐ Apenas projetos individuais
- ☐ Pequenas equipes (Menos de 4 pessoas)
- ☐ Divisão Departamental seguindo os moldes de uma organização hierárquica
- ☐ Divisão Departamental seguindo os moldes de uma organização matricial
- ☐ Organizada de forma ambígua ou incerta
- ☐ Outro(s) \_\_\_\_\_

**8. Quais são os objetivos de negócio na organização em relação à área de software?**

- ☐ Apenas comercialização de software
- ☐ Desenvolvimento de software para controle de atividades internas da empresa
- ☐ Desenvolvimento de software para ser incorporado em produto desenvolvido pela empresa
- ☐ Adequar os produtos da empresa a novas necessidades do mercado
- ☐ Outro(s) \_\_\_\_\_



### 6.3 Questionário de Projetos

1. Qual o tamanho da equipe de desenvolvimento? \_\_\_\_\_ Homens/mês

2. Qual o tempo de duração do projeto? \_\_\_\_\_ meses

3. Qual é a experiência da equipe?

- ☐ Pouca: a equipe possui apenas profissionais com pouca (de 2 a 5 anos) ou nenhuma experiência (menos de 2 anos)
- ☐ Média para pouca: grande parte da equipe possui pouca ou nenhuma experiência e há alguns profissionais experientes na equipe (mais de 5 anos)
- ☐ Média: mistura entre profissionais experientes e com pouca ou nenhuma experiência
- ☐ Média para muita: maioria dos desenvolvedores possui grande experiência na área
- ☐ Muita: apenas profissionais com grande experiência

4. Quais são os equipamentos/plataformas alvo e de desenvolvimento?

<b><i>Alvo</i></b>	<b><i>Desenvolvimento</i></b>
<input type="checkbox"/> Mainframe	<input type="checkbox"/> Mainframe
<input type="checkbox"/> Workstations	<input type="checkbox"/> Workstations
<input type="checkbox"/> PC	<input type="checkbox"/> PC
<input type="checkbox"/> Processadores Embutidos	<input type="checkbox"/> Processadores Embutidos
<input type="checkbox"/> Macintosh	<input type="checkbox"/> Macintosh
<input type="checkbox"/> Outro(s) _____	<input type="checkbox"/> Outro(s) _____

5. Quais áreas computacionais têm forte relação ao projeto e produto desenvolvidos?

- |  |  |
|--|--|
| <input type="checkbox"/> Internet                | <input type="checkbox"/> e-commerce            |
| <input type="checkbox"/> Multimídia              | <input type="checkbox"/> Banco de dados        |
| <input type="checkbox"/> Inteligência artificial | <input type="checkbox"/> Tempo real            |
| <input type="checkbox"/> Gráficos                | <input type="checkbox"/> Redes de computadores |
| <input type="checkbox"/> Automação e controle    | <input type="checkbox"/> Sistema operacional   |
| <input type="checkbox"/> Outra(s) _____          |  |

**6. Qual é a natureza do projeto?**

- ☐ Projeto Novo
- ☐ Melhoramento (acréscimo de novas funcionalidades)
- ☐ Manutenção (reparo de defeitos)
- ☐ Conversão ou adaptação (migração para outras plataformas)
- ☐ Outra \_\_\_\_\_

**7. Qual é a classe do projeto?**

- ☐ Pessoal
- ☐ Acadêmico
- ☐ Interno à organização (para ser incorporado em um dos produtos da companhia)
- ☐ Externo (para ser vendido como um dos produtos da companhia)
- ☐ Customizado (para ser desenvolvido para um cliente específico)
- ☐ Outra \_\_\_\_\_

**8. Qual é o tipo de projeto?**

- ☐ Não procedural (Query, datasheet...)
- ☐ Aplicação em Lote (Batch)
- ☐ Aplicação baseada em Banco de Dados
- ☐ Programa Interativo
- ☐ Programa Científico ou Matemático
- ☐ Sistema ou Programa de Suporte
- ☐ Programa de Tempo Real ou Embutido
- ☐ Programa de Inteligência Artificial
- ☐ Programa de Controle de Processos
- ☐ Programa de Animação ou Gráfico
- ☐ Programa de Comunicações ou Telecomunicações
- ☐ Robótica ou Programa de Automação Mecânica
- ☐ Outro \_\_\_\_\_

**9. Quais são os aspectos críticos de qualidade durante o desenvolvimento do projeto?**

- |   |   |
|---|---|
| <input type="checkbox"/> Funcionalidade           | <input type="checkbox"/> Manutenibilidade |
| <input type="checkbox"/> Confiabilidade           | <input type="checkbox"/> Portabilidade    |
| <input type="checkbox"/> Usabilidade              | <input type="checkbox"/> Outro(s) _____   |
| <input type="checkbox"/> Desempenho (performance) |   |

**10. Quais são os fatores que têm maior influência durante o desenvolvimento do projeto?**

- |   |  |
|---|--|
| <input type="checkbox"/> Velocidade de comunicação de dados | <input type="checkbox"/> Processamento distribuído |
| <input type="checkbox"/> Performance/desempenho             | <input type="checkbox"/> Taxa de transações        |
| <input type="checkbox"/> Memória                            | <input type="checkbox"/> Mudanças nos requisitos   |
| <input type="checkbox"/> Complexidade de processamento      | <input type="checkbox"/> Facilidade de operação    |
| <input type="checkbox"/> Mudança de hardware/plataforma     | <input type="checkbox"/> Facilidade de instalação  |
| <input type="checkbox"/> Outro(s) _____                     |  |

**11. Qual é a complexidade do projeto?**

- ☐ Baixa (algoritmos e cálculos simples)
- ☐ Média (a maior parte dos algoritmos é simples e alguns deles são complexos)
- ☐ Média para alta (boa parte dos algoritmos requer cálculos complexos)
- ☐ Alta (a maior parte dos algoritmos possui cálculos complexos)

**12. Quais são as ferramentas utilizadas durante o projeto?**

- |  |  |
|--|--|
| <input type="checkbox"/> Analisador de código                                      | <input type="checkbox"/> Gerador de telas            |
| <input type="checkbox"/> Gerenciador de bibliotecas de módulos                     | <input type="checkbox"/> CASE                        |
| <input type="checkbox"/> Ferramenta de depuração (DEBUG)                           | <input type="checkbox"/> Gerenciador de configuração |
| <input type="checkbox"/> Dicionário de Dados                                       | <input type="checkbox"/> Gerenciador de documento    |
| <input type="checkbox"/> Documentador  | <input type="checkbox"/> Gerenciador de projetos     |
| <input type="checkbox"/> Gerador de código fonte                                   | <input type="checkbox"/> Modelador de base de dados  |
| <input type="checkbox"/> Gerador de dados de teste                                 | <input type="checkbox"/> Otimizador de código        |
| <input type="checkbox"/> Biblioteca de código fonte/componentes                    | <input type="checkbox"/> Modelador de base de dados  |
| <input type="checkbox"/> Ferramenta de desenvolvimento visual                      | <input type="checkbox"/> Otimizador de código        |
| <input type="checkbox"/> Gerador de entrada de dados                               | <input type="checkbox"/> Prototipador                |
| <input type="checkbox"/> Gerador de gráficos                                       | <input type="checkbox"/> Gerador de relatórios       |
| <input type="checkbox"/> Outra(s) _____  |  |
| <input type="checkbox"/> Não utiliza ferramentas automatizadas de apoio ao projeto |  |

**13. Qual é o modelo de ciclo de vida adotado?**

- |   |  |
|---|--|
| <input type="checkbox"/> Build and Fix (melhoramento iterativo) | <input type="checkbox"/> Waterfall (Cascata) |
| <input type="checkbox"/> Prototipagem                           | <input type="checkbox"/> Incremental         |
| <input type="checkbox"/> Espiral                                |  |
| <input type="checkbox"/> Outro _____                            |  |

**14. Qual é a metodologia de desenvolvimento adotada no projeto?**

- ☐ Estruturada
- ☐ Orientada a objetos
- ☐ Outra \_\_\_\_\_

**15. Quais métodos para determinação de requisitos são utilizados?**

- ☐ JAD (Join Application Development)
- ☐ Baseada em entrevistas, métodos formais
- ☐ Baseada em entrevistas, métodos informais
- ☐ Nenhuma forma de determinação de requisitos é necessária (requisitos claros).
- ☐ Outro(s) \_\_\_\_\_

**16. Qual método de método de análise e projeto de sistemas é utilizado?**

- ☐ Análise funcional
- ☐ Modelagem de dados e entidades
- ☐ Análise estruturada
- ☐ Análise orientada a objetos
- ☐ Análise informal
- ☐ Nenhuma análise é realizada
- ☐ Outro \_\_\_\_\_

**17. Qual linguagem de programação é utilizada no projeto?**

- |                                       |                                    |                                 |
|---------------------------------------|------------------------------------|---------------------------------|
| <input type="checkbox"/> C            | <input type="checkbox"/> C++       | <input type="checkbox"/> Pascal |
| <input type="checkbox"/> Fortran      | <input type="checkbox"/> 4GL       | <input type="checkbox"/> Basic  |
| <input type="checkbox"/> COBOL        | <input type="checkbox"/> Assembler | <input type="checkbox"/> Delphi |
| <input type="checkbox"/> Visual Basic | <input type="checkbox"/> Smalltalk | <input type="checkbox"/> Java   |
| <input type="checkbox"/> Outra _____  |                                    |                                 |

**18. Quais métodos de teste e detecção de defeitos são empregados?**

- |  |  |
|--|--|
| <input type="checkbox"/> Auditorias            | <input type="checkbox"/> Testes de usabilidade |
| <input type="checkbox"/> Inspeções formais     | <input type="checkbox"/> Testes estruturais    |
| <input type="checkbox"/> Revisões estruturadas | <input type="checkbox"/> Testes funcionais     |
| <input type="checkbox"/> Testes de aceitação   | <input type="checkbox"/> Validação             |
| <input type="checkbox"/> Testes de campo       | <input type="checkbox"/> Testes de sistema     |
| <input type="checkbox"/> Testes de unidade     |  |
| <input type="checkbox"/> Outro(s) _____        |  |

**19. Quais tipos de documentação são adotados?**

- |   |  |
|---|--|
| <input type="checkbox"/> Contratos e acordos                  | <input type="checkbox"/> Documentação comercial    |
| <input type="checkbox"/> Resultado de revisões / testes       | <input type="checkbox"/> Documentação de marketing |
| <input type="checkbox"/> Documentação de programas            | <input type="checkbox"/> Especificação do sistema  |
| <input type="checkbox"/> Documentação do processo de software | <input type="checkbox"/> Manuais                   |
| <input type="checkbox"/> Documentação de descrição do produto | <input type="checkbox"/> Plano de testes           |
| <input type="checkbox"/> Projeto de sistema                   |  |
| <input type="checkbox"/> Plano de controle da qualidade       |  |
| <input type="checkbox"/> Outro(s) _____                       |  |
| <input type="checkbox"/> Não adota documentação               |  |

**20. Existe um grupo de garantia de qualidade (GQ) atuando no projeto?**

- ☐ Sim
- ☐ Não (*neste caso as perguntas posteriores não precisam ser respondidas*)

**21. Quais métodos de melhoramento de processos são utilizados no projeto?**

- ☐ ISO 9000
- ☐ CMM
- ☐ QIP
- ☐ Outro(s) \_\_\_\_\_
- ☐ Não possui nenhum método para melhoramento do processo

**22. Qual é a experiência do grupo de garantia de qualidade com respeito à mensuração?**

- ☐ Mais de 5 programas
- ☐ 1 – 5 programas realizados
- ☐ 1 programa
- ☐ Nenhum

**23. Como são realizadas as atividades do grupo de garantia de qualidade?**

- ☐ Atividades são planejadas e executadas de maneira formal
- ☐ Atividades são realizadas informalmente
- ☐ Não existem atividades de garantia de qualidade

**24. Existem medidas padrão de qualidade?**

- ☐ Sim
- ☐ Não

## 6.4 Análise e Projeto da Ferramenta

### 1. Análise:

**1.1. Escopo:** desenvolvimento de uma ferramenta que automatize o processo de seleção de índices e auxilie a utilização da abordagem de indexação proposta.

#### **1.2. Características necessárias:**

- Fornecer uma ferramenta de fácil utilização;
- Disponibilizar uma forma para o armazenamento da meta e dos casos coletados;
- Fornecer ao usuário um conjunto final de índices para os seus casos, além de permitir a alteração deste conjunto final;
- Implementar procedimentos/recursos que permitam o monitoramento e o aperfeiçoamento do conjunto de índices inicialmente proposto.

**1.3. Contexto:** o contexto do problema compreende a fase de identificação de características de contexto (organizacionais e de projeto) dentro do domínio da reutilização de experiências de software. A ferramenta irá auxiliar as fases de coleta e interpretação de dados adquiridos.

### 2. Projeto:

**2.1. Fatores de sucesso:** identificar índices que apresentem boa representatividade da base de experiência coletada e possuir procedimentos de aperfeiçoamento e *feedback* para revisar ou validar os índices identificados.

**2.2. Riscos:** pouco conhecimento sobre a linguagem Smalltalk e a ferramenta de desenvolvimento VisualWorks.

#### **2.3. Papéis e Responsabilidades:**

- Patrocinador: Christiane Gresse von Wangenheim,
- Analista, projetista e desenvolvedor: Marcel Pacheco de Souza,
- Usuário Líder: empresa Complex Informática Ltda.

**2.4. Atores:** usuário da base de conhecimento, desenvolvedor, gerente de projeto, gerente da organização e engenheiro de conhecimento.

Ator	Definição
Usuário da base de conhecimento	Utiliza o conjunto de índices que caracterizam a base de conhecimento.
Desenvolvedor	Fornecer dados sobre os projetos desenvolvidos ou em desenvolvimento.
Gerente de Projeto	Fornecer dados sobre os projetos desenvolvidos ou em desenvolvimento.
Gerente da Organização	Fornecer dados sobre a organização.
Engenheiro de Conhecimento	Valida os dados gerados nas fases da abordagem.

## 2.5. Lista de Eventos:

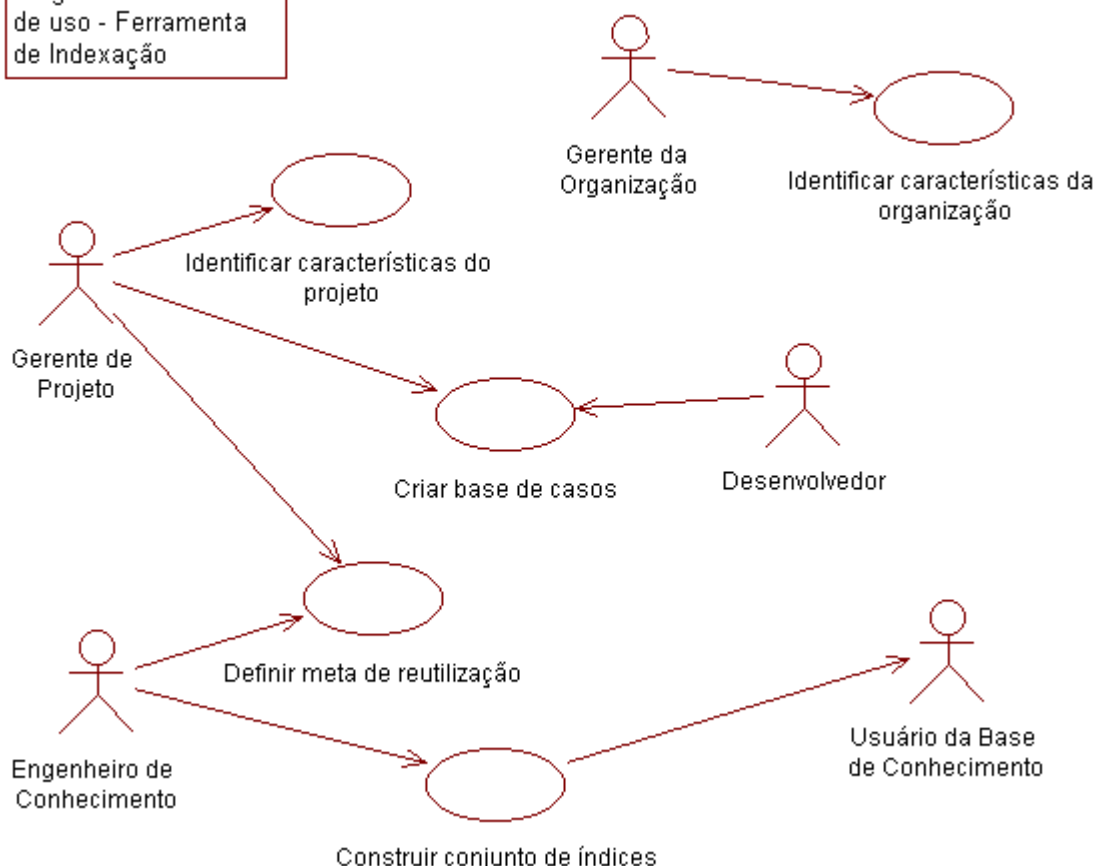
- Usuário da base de conhecimento + adquire + conjunto de índices
- Desenvolvedor + informa + caso
- Desenvolvedor + informa + característica de projeto
- Gerente da Organização + informa + característica da organização
- Gerente de Projeto + informa + característica de projeto
- Gerente de Projeto + informa + meta de reutilização
- Engenheiro de Conhecimento + valida + meta de reutilização
- Engenheiro de Conhecimento + valida + índices preliminares
- Engenheiro de Conhecimento + recebe + feedback
- Engenheiro de Conhecimento + redefine + conjunto de índices

Sujeito	Verbo	Objeto	Resposta
Usuário da base de Conhecimento	adquire	Conjunto de Índices	O conjunto de índices é gerado e disponibilizado.
Desenvolvedor	informa	Caso	Caso é colocado na base de casos avaliáveis.
Desenvolvedor	informa	Característica de Projeto	Característica de projeto é definida.
Gerente da Organização	Informa	Característica da Organização	Característica da organização é definida.
Gerente de Projeto	informa	Característica de Projeto	Característica de projeto é definida.
Gerente de Projeto	informa	Metas de Reutilização	Meta de reutilização é definida.
Engenheiro de Conhecimento	valida	Metas de Reutilização	Meta de reutilização é analisada, ajustada e definida.
Engenheiro de Conhecimento	valida	Índices Preliminares	Índices são analisados, revisados e definidos.
Engenheiro de Conhecimento	recebe	Feedback	Informações referentes ao uso dos índices são fornecidas.
Engenheiro de Conhecimento	redefine	Conjunto de Índices	Conjunto de índices é redefinido baseado nos dados de feedback gerados.



## 2.6. Casos de Uso:

Diagrama de caso de uso - Ferramenta de Indexação



## 2.7. Curso de Eventos de Casos de Uso:

Nome do caso de uso	1 - Definir meta de reutilização
Descrição do caso de uso	O(s) gerente(s) de projeto com o apoio de um engenheiro de conhecimento deverão chegar a um conjunto de metas de reutilização para construção de uma base de conhecimento dos produtos dos projetos realizados pela organização.
Atores	Gerente(s) de projeto e engenheiro de conhecimento
Pré-condições	O(s) gerente de projeto deverão ter comandado ou ter conhecimento a respeito dos projetos realizados pela organização. O engenheiro de conhecimento deve possuir conhecimento sobre o método de indexação proposto e sobre reutilização de conhecimento.
Produto	Conjunto de metas de reutilização.

<b>Nome do caso de uso</b>	<b>2 – Identificar características da organização</b>
Descrição do caso de uso	Através de uma pesquisa com o gerente da organização dados referentes à organização são coletados e armazenados. Estes dados serão analisados posteriormente.
Atores	Gerente da organização
Pré-condições	O gerente da organização conhece as características da empresa, sua forma de organização, tamanho da equipe de desenvolvedores, linha de projetos desenvolvidos...
Produto	Um conjunto de características organizacionais é definido.

<b>Nome do caso de uso</b>	<b>3 - Identificar características do projeto</b>
Descrição do caso de uso	Através de uma pesquisa com o(s) gerente(s) de projeto dados referentes a projetos realizados e em andamento são coletados e armazenados. Estes dados serão posteriormente analisados.
Atores	Gerente de projeto
Pré-condições	O gerente de projeto está gerenciando ou gerenciou um projeto realizado pela organização.
Produto	Um conjunto de características de projeto é obtido.

<b>Nome do caso de uso</b>	<b>4 - Criar base de casos</b>
Descrição do caso de uso	Participantes de projetos de desenvolvimento de software informam algumas experiências durante o processo de construção do software.
Atores	Desenvolvedor
Pré-condições	A equipe de desenvolvedores deve ter conhecimento prático sobre o desenvolvimento de software.
Produto	Uma base de casos é obtida.

<b>Nome do caso de uso</b>	<b>5 – Construir conjunto de índices</b>
Descrição do caso de uso	Gerar um conjunto de indexadores para as características organizacionais e de projeto da base de experiências de desenvolvimento de software da organização.
Atores	Usuários da base casos e engenheiro de conhecimento
Pré-condições	As metas de reutilização, o conjunto de características organizacionais e o conjunto de características de projetos realizados pela organização já deverão estar disponíveis e catalogados.
Produto	Um conjunto de características indexadoras é definido, juntamente com um conjunto de possíveis valores para cada característica.

## 2.8. Arquitetura Preliminar:

Componente	Implementação
Hardware	Mínimo: Pentium 100Mhz com 32MB de RAM e disco rígido de 500MB Ideal: Pentium III 800Mhz com 64MB de RAM e disco rígido de 2GB
Software Sistema Operacional	Windows
Software de desenvolvimento	VisualWorks
Software de modelagem visual	Rational Rose

## 2.9. Modelo de Classes:

